D2.3 Common Data Models and Semantic Interoperability Mechanisms - Release 2

Dissemination Level: Public
Submission Date: 30/04/2021

Table of Contents

List of Figures

European Union
European Regional
Development Fund

List of Tables

1 Executive Summary

DEMETER aims to lead the Digital Transformation of the European Agrifood sector based on the rapid adoption of advanced technologies, such as Internet of Things, Artificial Intelligence, Big Data, Decision Support, Benchmarking, Earth Observation, etc., in order to increase performance in multiple aspects of farming operations, as well as to assure the viability and sustainability of the sector in the long term. It aims to put these digital technologies at the service of farmers using a human-in-the-loop approach that constantly focuses on mixing human knowledge and expertise with digital information. DEMETER focuses on interoperability as the main digital enabler, extending the coverage of interoperability across data, platforms, services, applications and online intelligence, as well as human knowledge, and the implementation of interoperability by connecting farmers and advisors with providers of ICT solutions and machinery.

To enable the achievement of the aforementioned objectives, and to promote the targeted technological, business, adoption and socio-economic impacts, DEMETER has already delivered a revised and updated Reference Architecture (RA) that is suitable to address these challenges in the agrifood domain. One of the key technologies that is necessary to implement this Reference Architecture and arguably the most crucial of these is the common data models which make the DEMETER Agricultural Information Model (AIM) and which enable semantic interoperability between DEMETER and existing agrifood systems and ontologies. This deliverable introduces the second updated release of the DEMETER common data models and interoperability mechanisms which have been updated based on the experience and the lessons learned from the implementation of the DEMETER enablers and tools for the first round of pilots. This update took place following a specific methodology, where information was collected by the pilots and developers regarding changes to the AIM model, so as to enabler missing functionality and new semantic mappings. After presenting this methodology, we present the findings for questionnaires given to the project partners regarding AIM changes, and then present an updated view of the technical requirements that drive the development of the DEMETER AIM. It then describes in detail the revised design and development of the DEMETER Agricultural Information Model and its core metamodel; next, the cross-domain ontology, followed by the domain specific ontologies and pilot specific ontologies; and, finally, the AIM metadata schema. Following this description, it discusses the interoperability support between the DEMETER AIM and several existing ontologies and dominant agri-food systems by presenting the updated semantic mapping of those to AIM. Afterwards, it presents the updated AIM implementation which still follows a layered and modular approach, reusing as much as possible existing ontologies and vocabularies. Subsequently, the current state of usage of AIM across the pilots is presented focusing in particular on elements such as AIM usage guidelines for pilot developers, and approaches for developing data AIM wrappers, as well as further AIM extensions and pilot-specific ontologies that are requested by pilots. Finally, the document concludes presenting also future work towards the final version of the AIM which will be driven by the requrests from pilots and partners and a further review or state of the art ontologies that can be aligned to AIM.

European Union
European Regional
Development Fund

2 Acronyms

ADAPT	Agricultural Data Application Programming Toolkit
AIM	Agricultural Information Model
AKIS	Agricultural Knowledge and Innovation Systems
API	Application Programming Interface
CAN-Bus	Controller Area Networks Bus
CEFACT	Centre for Trade Facilitation and E-Business
CIM	Context Information Management
CO2	Carbon Dioxide
DCAT	Data Catalog Vocabulary
DQV	Data Quality Vocabulary
DSS	Decision Support System
DUV	Data Usage Vocabulary
EO	Earth Observation
EPPO	European and Mediterranean Plant Protection Organization
ETSI	European Telecommunications Standards Institute
EU	European Union
EVI	Enhanced Vegetation Index
FAIR	Findable, Accessible, Interoperable and Reusable
FAO	Food and Agriculture Organization
FMIS	Farm Management Information System
GPS	Global Positioning System
GSMA	Global System Mobile Association
GTIN	Global Trade Item Number
HTTP	HyperText Transfer Protocol
ICT	Information and communications technology
IETF	Internet Engineering Task Force
IoT	Internet of Things
ISG	Industry Specification Group

ISO	International Organization for Standardisation
JSON	Java Script Object Notation
JSON-LD	Java Script Object Notation - Linked Data
KPI	Key Performance Indicator
M2M	Machine-to-Machine
NCDX	Nordic Cattle Database Exchange
NDMI	Normalized Difference Moisture Index
NDRE	Normalized Difference Red Edge
NDVI	Normalized Difference Vegetation Index
NGSI	Next Generation Sensors Initiative
NGSI-LD	Next Generation Sensors Initiative - Linked Data
OBO	Open Biological and Biomedical Ontology
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
OWL-RL	Web Ontology Rule Language
PROV-O	The Provenance Ontology
pySHACL	Python Shapes Constraint Language
QUDT	Quantities, Units, Dimensions, and Types Ontology
RA	Reference Architecture
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SAREF	Smart Appliances REFerence (SAREF) ontology
Saref4Agri	Smart Appliances REFerence (SAREF) ontology for agriculture
SHACL	Shapes Constraint Language
SKOS	Simple Knowledge Organization System
SKOS-XL	Simple Knowledge Organization System eXtension for Labels
SOSA	Sensor, Observation, Sample, and Actuator
SPARQL	Simple Protocol and RDF Query Language
SW	SoftWare

SSN	Semantic Sensor Network
SWRL	Semantic Web Rule Language
TTL	Turtle (file)
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
URI	Uniform Resource Identifier
UN	United Nations
URL	Uniform Resource Locator
US	United States
W3C	World Wide Web Consortium
WGS84	World Geodetic System 1984
WoT	Web of Things
WP	Work Package
XML	Extensible Markup Language

3 List of Authors and Reviewers

Organisation	Author
ICCS	Ioanna Roussaki (Editor)
	Ioannis Vetsikas
	George Routis
	Marios Paraskevopoulos
PSNC	Raul Palma
	Soumya Brahma
	Szymon Mueller
Fraunhofer FIT	Till Döhmen
	Christoph Lange-Bever
OGCE	Rob Atkinson
Fraunhofer IESE	Anna-Maria Volmer
TECNALIA	Belén Martínez
	Alejandro Rodríguez
	Fernando Jorge
ENG	Antonio Caruso

Organisation	Reviewer
OdinS	Juan Antonio Martinez Navarro
TECNALIA	Sonia Bilbao

4 Introduction

This deliverable presents the second release of the "DEMETER Common Data Models and Semantic Interoperability Mechanisms". It mostly presents the revision of the DEMETER Agricultural Information Model (AIM) which is the data model used by all partners for the first round of the DEMETER pilots. The changes are underlined by the revised technical requirements, new needs identified when trying to deploy and integrate pilots to AIM and, of course, updates and new approaches in State of the Art.

More specifically, the rest of the document is structured as follows:

Section 5 provides an analysis of the methodology followed to identify the required updates on AIM and the subsequent revisions on the model. Specifically, there are references about the current technological trends as presented in State of the Art, the revised pilot requirements captured as input by the respective questionnaires, the use of Issue tracker as complementary tool for capturing pilot or developer needs for AIM updates and a brief description of the changes.

Section 6 provides with an overview of the findings from the aforementioned questionnaires and the way they contributed to the revision.

Section 7 gives an overview of the revised technical requirements. This is an exhaustive list of specific technical requirements that the DEMETER Agricultural Information Model needs to be able to represent, as well as requirements regarding the interoperability with existing systems and ontologies, including the mapping of these data models to the DEMETER AIM. The requirements are revised compared with those in the initial release; these revisions are a result of the AIM implementation and the analysis performed (which is described in Section 5).

Section 8 presents the revision of the DEMETER Agricultural Information Model (AIM) in detail that follows a modular approach in a layered architecture for its development. More specifically:

- subsection 8.1 presents the AIM core metamodel, which follows the NGSI-LD [ETSI19] meta-modeling approach;
- subsection 8.2 presents the revision of the AIM cross-domain ontology used, i.e., the set of generic models which aim at providing common definitions for all agrifood domain handled by the AIM and at avoiding conflicting or redundant definitions of the same classes at the domain-specific layer;
- subsection 8.3 presents the revised AIM domain-specific ontologies developed for the AIM, which model information such as crops, animals, agricultural products as well as farms and farmers, etc;
- subsection 8.4 presents the newly-added pilot-specific ontologies layer. The layer comprises of tailor-made ontologies to our pilots which extend ontologies from the upper level and mostly define new concepts that exist in no other known agriculture ontology;
- Subsection 8.5 describes the revised metadata schema used by AIM. It expresses semantics, related to meta-information about the datasets, based on the cross-domain and domain specific ontologies previously presented.

Section 9 presents the interoperability support between the DEMETER AIM and several existing ontologies and dominant agri-food systems detailing the revised semantic mapping of these to the AIM. More specifically, it presents this information for the following dominant systems: FIWARE AgriFood, Saref4Agri, ADAPT, INSPIRE (and FOODIE), AGROVOC and EO data. There is also presented a new semantic mapping to EPPO.

Section 10 presents the implementation of the DEMETER AIM which follows a layered and modular approach, reusing as much as possible existing ontologies and vocabularies. More specifically, it presents the implementations for the different layers (parts) of AIM, together with the design and implementation choices taken, the mappings implemented and the tools used during the implementation process.

Section 11 explains the current state of usage of AIM across the pilots. Specifically, the following aspects are covered: AIM usage guidelines for pilot developers, pilot existing approaches for data modelling/semantics and the respective AIM wrappers, the datatypes modelled via AIM across the pilots and further AIM extensions and pilot-specific ontologies that are requested by pilots and shall be implemented for the next release.

Finally, Section 12 concludes the document also presenting work towards the foreseen semantic mapping of AIM to more ontologies and future work to expand AIM, while Section 13 provides the respective references used.

The Annexes of this document provide additional information regarding the future work on AIM revisions escoreted by an extension of the D2.1 state of the art review and the full questionnaire responses. Moreover, to keep this document self-contained there are two more Annexes included providing the overall detailed description of the AIM core meta-model and the domain specific ontologies layers, as the first one being presented only in summary in Section 8, while for the later only the revisions made are highlighted in the same Section.

Moreover, since in the body of this deliverable only a brief summary of the AIM core meta-model is presented, while only the revisions made over the domain specific ontologies layer are highlighted (both in Section 8), in order to keep this document self-contained, there are two more Annexes appended providing the overall detailed descriptions of these two AIM layers.

The models and interoperability mechanisms presented in this deliverable complement the deliverable D3.3 DEMETER Reference Architecture (Release 2). Moreover, the work presented herewith will have an impact on a number of forthcoming deliverables, i.e.:

- D2.4 DEMETER data and knowledge extraction tools – Release 2 (May 2021)
- D3.4 DEMETER technology integration tools – Release 2 (June 2021)
- D4.3 Decision Support, Benchmarking and Performance Indicator Monitoring Tools – Release 2 (May 2021)
- D4.4 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space - Release 2 (June 2021)
- D5.6 Testbed, deployment, system extensions and applications for pilot round 2 (August 2021)

The final release of the DEMETER Common Data Models and Semantic Interoperability Mechanisms is planned for release on October 2022 and will be presented in Deliverable 2.5.

5 Methodology for and overview of AIM Revisions

The evolution of the DEMETER pilots and the integration of various tools and enablers into them, natuarally results in the identification of additional concepts that need to be represented via the AIM model, or of necessary revisions over existing model elements or requirements in order to ensure model completeness, efficiency and semantic interoperability.

In this section, the methodology used in order to extract the necessary AIM revisions is initially presented. This includes for example the usage of questionnaires, establishment of an issue tracker in order track changes over the AIM model, as well as direct input from the technical work package developers and partners. Following this, a summary of the extensions and revisions that were made over the DEMETER AIM model is provided. The goal is to present an overview of what will be elaborated upon in the next sections of this deliverable and also to summarize the key AIM changes since its initial release described in Deliverable 2.1.

5.1 Methodology for AIM revisions in Release 2

Regarding the methodology used in order to identify necessary AIM revisions, firstly, we needed to build upon the experiences derived from the implementation of the various DEMETER tools and the whole system as well as the development and usage of AIM in the various enablers developed and used during the first round of pilots and the experience with using these components and the instantiations to the various pilot sites and applications. In this respect, a number of methods have been used in order to get the necessary information regarding the extension needed to AIM.

The most important of these is undoubtedly the issue tracker established on GitLab, where the code for AIM repository and several of the DEMETER core enablers and tools may be found and which is used to support the continuous integration and development of these. This tool has allowed all DEMETER partnera and developera to give immediate feedback and requests to the AIM developers regarding missing concepts and necessary updates (or sometimes corrections) and this has speed up significantly the reactions and turnaround time of the AIM revisions needed. A screenshot of the issue tracker is provided in Figure 1.

The issue tracker is a project that adjusts the corresponding feature offered by GitLab and aims to facilitate the documentation of bugs, suggestions or/and questions regarding Work Package 2. The concept is that someone can add an issue on the board or the list that corresponds to the board, both can be found via the link[1] or via the left-side menu on WP2 project on GitLab. There, anyone may record needed AIM revisions, comment on the model and suggest updates or even discuss various concepts. The developers that are responsible for AIM then identify the needs recorded on the tracker issues, discuss the optimal implementation over comments and then work towards satisfying the needs on the matter. The issue tracker has been instrumental in the extension of AIM and to speedily push through AIM revisions as needed by the DEMETER partners and pilots.

[1] https://gitlab.com/groups/demeterproject/wp2/-/boards

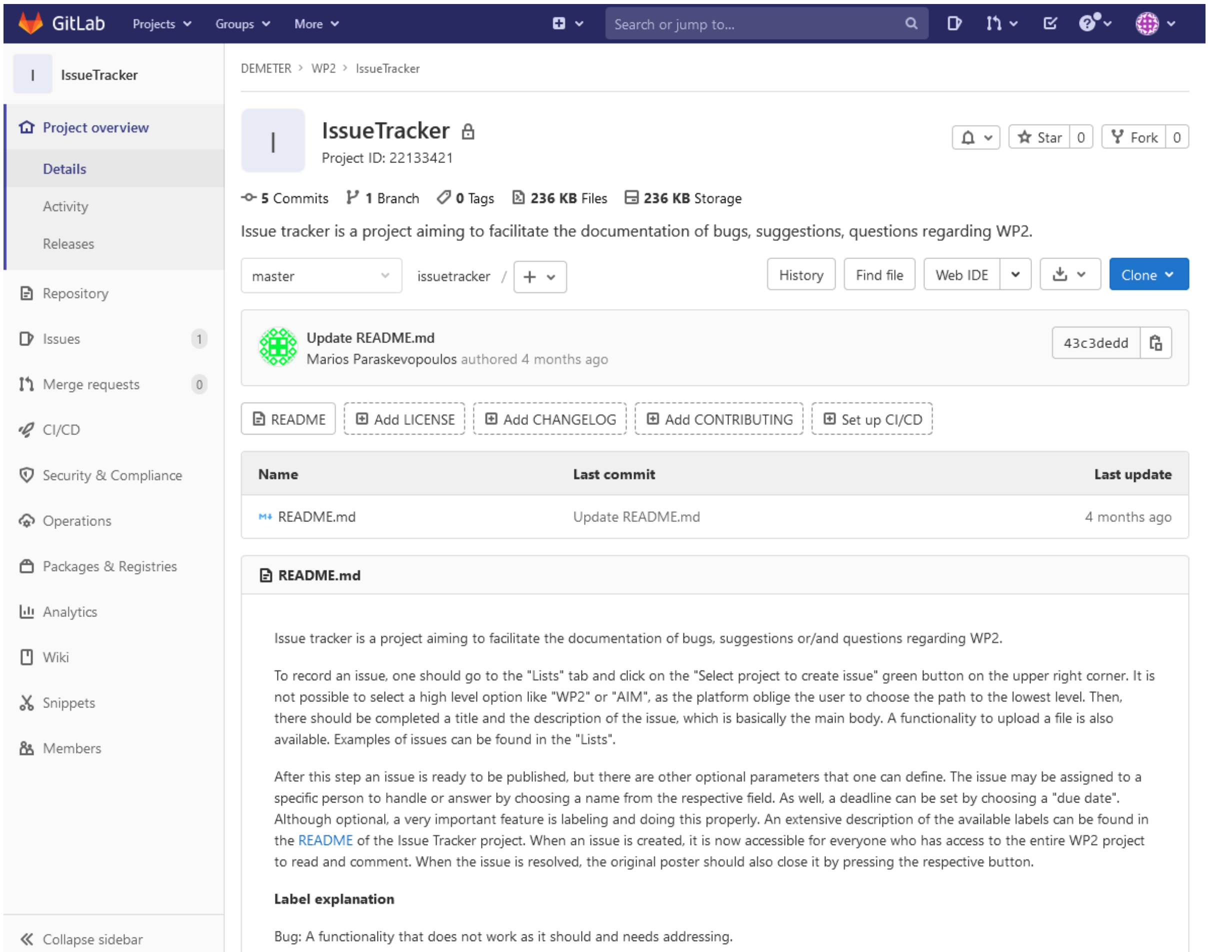

Figure 1. AIM (WP2) issue tracker

Secondly, we have collected and analyzed the responses to questionnaires between the conclusion of the first round of pilot implementations and the finalization of this deliverable, in order to receive input from the pilot users (and the pilot developers) regarding any changes and extensions to the AIM model, mostly due to missing concepts. These are presented in Section 7 of this deliverable and they were useful to identify any issues which might have not been already raised by the issue tracker (although indeed most were already raised and concluded using the issue tracker). Similarly, we also gave questionnaires and continuously queried the technical partners for WP2, 3 and 4 of DEMETER who were developing the enablers used in the pilots for feedback regarding potential revisions to AIM both to address issues such as missing concepts, but also to improve AIM in general (e.g., as part of a revision several concepts common to many domain specific ontologies of AIM were moved to the cross domain ontology to optimize the structure of the AIM ontology). More details on this are provided in Section 6.

Thirdly, we re-examined the technical requirements initially extracted by Deliverable 2.1 to determine whether AIM and the enablers and tools developed covered the requirements and whether some changes were needed. Additionally, to determine whether the requirements pertain to all the pilots and enablers or to specific pilots only a thorough study has been made. All this information is presented in Section 7 together with the full updated (where needed) text of the technical requirements for AIM.

Fourthly, building upon the feedback of the DEMETER partners and developers, we also examined some additional ontologies and systems which could be included into the AIM ontology. These are summarized in the annex as potential extensions to be addressed by the final version of AIM. Although their integration has not been concluded yet (in fact in some cases it has not even started) this review of other existing well-known ontologies that could be useful to be integrated within AIM and to provide semantic interoperability with them is another driver of the planned updates and also part of the methodology we used in order to decide on these updates.

Finally, the evolution of the related state of the art has been studied. In this respect, additional concepts or changes have been identified across all AIM layers that have eventually been implemented in the second AIM release. Moreover, there are several potential revisions identified as future work that will be investigated for implementation in the final AIM release to be delivered in October 2023. The analysis of the respective state of the art work, along with the potential identified changes and mappings are provided upon in Annex A.

5.2 Overview of AIM revisions in Release 2

Having presented the methodology for the AIM revisions, in this subsection the key changes that were performed to revise the AIM model and which are described in more detail in the various sections of this deliverable are summarized.

To begin documenting the changes, the AIM model adopts a layered approach (core meta-model, cross domain ontology, etc.) and this has not changed since its initial release captured in Deliverable 2.1. However, a new layer has been introduced in the second AIM release, i.e., the pilot-specific ontologies layer, which builds upon the domain-specific ontologies and aims to capture and add concepts and terms to AIM which are useful for specific pilots. Furthermore, most of these layers have changed to some degree, with the exception of the core meta-model which did not have any major change; the main implementation change to this meta-model which still uses json-ld (ngsi-ld), is that the json-ld context was updated in order to use prefix “meta”(e.g., meta.Property) to avoid naming conflicts with other cross or domain layer elements and the removal of the mapping of “id” and “type” to json-ld keys also to avoid conflicts in other layers.

On the other hand, the cross-domain ontology changed substantially. The current implementation integrates the relevant elements from different general standard/well known vocabularies including W3C/OGC [OGC16] standard SOSA/SSN, OGC Geosparql (including the simple_features_geometries extension), W3C RDF data cube, QUDT (plus the related QU ontology and extensions), W3C OWL Time ontology, plus other common vocabularies like FOAF, Dublin Core and Schema.org; any elements of these, which previously existed in domain specific ontologies, have therefore been moved here.

Moreover, some of the domain-specific ontologies have been updated driven by pilot needs and based on requests submitted (mostly) via the issue tracker. Some general terms have been moved to the domain specific ontologies. In many cases, additional terms have been added in the pilot-specific extensions, the aforementioned newly introduced layer that works closely with the domain-specific ontologies.

The AIM Metadata Schema has also undergone a major revision, in order to properly reflect IDS, as well as DCAT and DQV references. It maintains the original URI, but the previous Metadata Schema is replaced with an entirely new version. It builds upon DCAT 2.0 and uses a subset of the Industrial Data Space (IDS) Information Model to increase the expressivity of certain aspects of the AIM Metadata Schema.

The full details of the updates over the AIM model and all its layers are documented in Section 8, and more specifically in subsections 8.1, 8.2, 8.3, 8.4 and 8.5, the core meta model, the cross domain ontology, the domain-specific ontologies, the pilot-specific ontologies and the metadata schema are presented respectively.

Finally, the semantic mappings of AIM to various well-known ontologies and systems have also been revised since the initial AIM release captured in Deliverable 2.1. In this respect, several changes have been introduced to the semantic mappings of AIM to FIWARE, SAREF4AGRI, ADAPT, INSPIRE/FOODIE and AGROVOC. Furthermore, there is a new AIM mapping introduced regarding the EPPO[2] modelling approach. The aforementioned revised semantic mappings of AIM are presented in Section 9.

[2] European and Mediterranean Plant Protection Organization

6 Overview of questionnaire findings

This section presents the findings obtained from the questionnaires that were distributed to all DEMETER partners in order to identify missing concepts and extensions needed to be added to the AIM model.

The questionnaires that were distributed and processed had Excel format, and the following questions were included:

Questionnaire about revisions and extensions to AIM

- Type of AIM revision requested/implemented with the following possible answers:
  - change to Core Meta Model
  - change to Cross-domain ontology
  - change to Domain Specific ontologies
  - change to Meta Data Schema
  - alignment with existing ontology
  - new extension.
- Issue Reporting Date: where the respondent either fills in the form's submission date or the date of the issue being reported on the issue tracker.
- AIM Revision details: indicate which ontology to align with or the missing concepts that are requested to be added to AIM.
- Revised SW Gitlab link: url to the gitlab repository relevant to request.
- Related Issue link: the relevant link to the WP2 Issue Tracker, if the issue was also reported there.
- Related to Pilot X.Y or to Task x.y: insert the task or pilot that will use the new/revised data concepts.
- Requested by: where they should fill in the name of the person filing the request and the (short partner name).
- Contact Person(s): where they should fill in the WP2 developer handling the implementation of the requested AIM revision.
- Status of AIM revision with the following possible answers:
  - proposed
  - pending
  - reviewing
  - concluded/finalized
- Other Comments: for any other relevant comment that should be taken into account.

We have merged the answers to all questionnaires in a table that is presented in Annex B. Subsequently we have compiled a summary of the revisions requested where these are divided into six areas, depending on whether the type of revision relates to changes to the **Core Meta Model**, to the **Cross-domain ontology**, to the **Domain Specific ontologies**, or to **Meta Data Schema**, or it pertains to a new **alignment with an existing ontology**, or finally it is a completely **new extension**. In some cases (i.e., changes to the core meta-model and to the metadata schema) no real revisions have been requested, nor implemented. For the rest of the types, we report the revision and the pilot (or task) it is related to.

1) Changes to Core Meta-Model:

There were no actual revisions suggested over the core meta model. One response only mentioned the core meta-model, and this was due to misinterpretation, as the request was in fact relevant to concepts for a specific pilot (changing the pilot farm for their pilot) and not an actual core meta-model revision request.

2) Changes to the Cross-domain ontology:

The changes and extensions to the cross-domain ontology were related primarily to Pilot 1.1/1.2. The following revision was requested:

- There is a pending status of AIM in the following missing properties: FIWARE AgriCrop, Agrifood; Saref4agri s4agri:Crop and s4agri:PlantGrowthStage, which stand for the crop missing properties.
- Soil missing properties, such as FIWARE AgriParcelRecord; Saref4agri s4agri are pending, irrigation missing properties such as Saref4agri s4agri:WateringSystem are pending.
- Also pending are missing Weather properties (proposal), such as related Agriculture datamodels: FIWARE WeatherObserved, Agrifood WeatherObserved and Forecast missing term and properties.
- There are also pending request regarding image properties necessary to represent an Image entity with an url to the image, etc and lastly pending weather information time series in order to represent a timeseries of weather information (multiple attributes as wind speed, sun radiation, etc.).

3) Changes to the Domain-Specific ontologies:

Several partners and pilots requested revisions here:

- The following missing concepts were requested: Leaf Wetness (not humidity on the air) is missing from one of the following ontologies: agriCommon.ttl, agriCrop.ttl, agriProduct.ttl, agriProfile.ttl.
- Custom attributes for Plant Stress Detection and Custom attributes for Nitrogen Balance Model.
- Change in Kpi Indicator extension, more specifically add to the Kpi a set of sub-sectors to better group togeher similar Indicators. He would like to maintain the sector object working as usual add a sub-

sector. For subsector he would use the SKOS ontology as done for the sectors.

- Changes/corrections were requested: in one of the examples, the validFrom and validTo fields are not in a standard yyyy-mm-dd data formats: a valid date string for all date types should be enforced.
- There was a wrong definition of parcel and harvest date inside AgriCrop, because the parcel has a reference to crop and not the opposite; this issue has since been dealt with and concluded/finalized.
- Define of a set of AgriCrop vocabulary to share among all pilots, and the issued is finalized/concluded.
- **D**efining an observation time series for a string value, and this has since been concluded/finalized.
- A need to express for a specific location (expressed as geo:Feature) a set of observation about the pest stage. The issue is also finalized/concluded.
- Another issue that was concluded/finalized, it was about a possible misunderstanding on elements AgriCrop and AgriParcel.
- An issue about AIM compliant time series proposal and it has been concluded/finalized.
- An issue about Fields not available in AIM, which has been concluded/finalized.

4) Changes to the Meta Data Schema:

No revisions have been requested over the AIM metadata schema.

5) Alignment with an existing ontology:

A couple of changes were requested here. More specifically, pilot 3.3 requested help with a change to the initial AIM model due to changes in one of their components, which is pending. Pilot 5.2 requested, and this has been finalized, the update of the SHACL validation tool in order to allow inference mechanisms which are required to accept Point as valid geometry of FeatureOfInterest.

6) New extensions:

This refers to new concepts that need to be added to AIM as they are needed by pilots and they are not in the original implementation. A number of requests has been made:

- Pilot 5.1 requested custom attributes for vehicle properties and driver behaviour; these have been concluded/finalized.
- Pilot 4.4 requested custom fields for silos and custom fields for poultry well-being; also concluded/finalized.
- Pilots 5.4 and 5.1 requested component custom fields for transport condition; this is concluded/finalized.
- Pilot 1.3 requested a FertilizerEstimation extension with multiple data properties needed e.g.,

nitrogenUptake, nitrogenConcentration, soilRatio, chlorofyllIndex, mcari (chlorofyllAbsorption). Furthermore, it has been noticed that more research is required in order to work whether more concepts are also missing and need to be added.

- Pilot 2.1 proposed the addition of the following concepts (which are related to machines): NOx in, NOx in Sensor, NOx out, NOx out Sensor, Charge Air Temperature, DEF Level, DEF Dosing Temperature, DPF Regeneration Status, Engine Coolant Temperature, Fuel Temperature, Engine Oil Temperature, Engine Oil Pressure, Crank Case Pressure, Fuel Delivery Pressure, Engine Coolant Pressure, Engine Coolant Level, Engine speed, SCR in Temperature, SCR out Temperature.
- Pilot 3.2 partners propose that they need to send GeoPNG/GeoTiff images (such as NDVI, LAI) and trap images inside JSON-LD/AIM using base64 encoding (in fact this is the only way to get it into AIM format), however this will reduce the number of ports to be open and reduce authentication procedures; this issue is reported concluded/finalized.
- Pilot 3.4 requests to include "predictedProductionAmount" in Foodie:productionAmount, and so instead of using measured yield, this field should allow them to enter a value for the yield that is predicted by an algorithm.
- In Pilot 4.2 they finalized the recommended inclusion and use of time series. They have also concluded/finalized the Implemented property collections.
- In the same Pilot they concluded/finalized the addition of new data properties in order to manage the prediction algorithm metrics.
- In Pilot 5.3, they have concluded/finalized the areaCultivated, the forecrops, the seedAmount, the soilResult (which have been added to AIM), the news, the issue. Some changes are pending though: the meteoId and the plantDensity. They also proposed to provide input/output data for milk yield prediction API and another possible extension related to milk event data from NCDX in order to facilitate data integration with NCDX.

What we can conclude from this analysis is that most of the requested changes have been to add missing concepts and to fix small issues with the AIM implementation rather than to add many concepts and align with existing ontologies for which AIM missed some necessary interoperability. For more details about the actual responses to the questionnaires please refer to Annex B, which compiles all the responses submitted by the DEMETER partners. This Annex provides a table carrying an overview of the overall responses to the AIM revision questionnaires, carrying all details received regarding the following: Type of AIM revision requested/ implemented, Issue Reporting Date, AIM Revision details, Related Issue link (in WP2 Issue Tracker in Gitlab), Related to Pilot x.y or to Task x.y, Requested by, Contact Person(s), Status of AIM revision, as well as any other comments the participants would like to highlight.

7 Revised Technical Requirements

This section presents the technical requirements as extracted by Task 2.1. The initial requirements have been captured in D2.1. These have then been updated based on the development of AIM and of the various enablers developed based on the AIM format. The original requirement template used has been extended with two more fields: one capturing the enabler(s)/module(s) addressing the respective requirement and another indicating the status of delivering the functionality resulting from the specific requirement.

In the rest of this section, the detailed description of the revised/extended technical requirements is provided, as these have been updated with regards to the originally extracted requirements by Task 2.1, which have been captured by D2.1. Moreover, for reasons of completeness, the requirement fields that remain unchanged are also provided. We also summarize in the following table the changes that have been performed to the technical requirements presented in Sections 7.1 and 7.2.

7.1 Data Model and Data Modelling

Requirement ID	DK1.1	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Common data view for heterogeneous models				
Description	DEMETER needs to define a common data model that defines a common view on all heterogeneous entities connected and all the data involved in the pilots. This common data model shall be used for all data exchanged between software components. Therefore, it needs to support the translation of the obtained data streams to a common data model.				
Addressed by Enabler(s) / Module(s)	AIM				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance existing (and if necessary introduce new) Information Models Objective 2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	8. Unified agriculture ontology				
Involved stakeholders/actors	Domain experts, pilot leaders, semantic technologies experts				

Prerequisite(s)	Pilots' requirements
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS, TECNALIA, PSNC

Requirement ID	DK1.3	**Version**	1.0	**Last Update Date**	09/04/2021
Title	Representation of crop farms data				
Description	DEMETER's data model needs to enable the common representation of agronomic data (e.g., crops, sensor data from the field, thermal/multispectral imagery from unmanned and autonomous vehicles (e.g., UAV, UGV), production data, geolocation data, planting data, irrigation logs, fertilisation logs, spraying logs, …) including: • Farm and economics modelling: agricultural type and economic size, production volumes and types, calculations according to results, etc. • Field data modelling: location and geometry of the field, planting date, planting distance, detailed yield information. • Field status modelling: e.g., water- or nitrogen-stressed fields, appropriate evaluation indices (e.g., Normalized Difference Moisture Index (NDMI)), need for fertilising. • Soil data modelling: soil temperature and moisture, soil physical and chemical analysis • Crops, treatment and fertilisation modelling: crop type, crop developing stages, crop cultivar or variety, crop health status and pests, pesticides, nitrogen levels, information from counting devices used for the control of insects or plagues. • Traceability information of crops (production, transport, retail) to be used in the product passport information • Water management modelling: water and energy consumption, water quality (e.g., salinity levels)				
Addressed by Enabler(s) / Module(s)	AIM (Agriculture Crops, Agriculture Interventions, Field Operations, Nutrient Monitor modules)				
Relevant Pilot(s)	• Farm and economics modelling: 2.4 • Field data modelling: 3.4				

	• Field status modelling: 1.3, 3.1, DK2.2 • Soil data modelling: 1.4, 3.2 • Crops, treatment and fertilisation modelling: 1.3, 1.4, 2.2, 3.1, 3.2, 3.3, DK2.1, DK2.3 • Traceability information of crops: DK2.1 • Water management modelling: 1.1, 1.2, 1.3, 1.4, 3.1, 3.2
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models
Relevant Innovation(s)	8: Unified agriculture ontology 17: Water Management Model
Involved stakeholders/actors	Solution providers, standardisation organisations
Prerequisite(s)	Data models should be based on existing ontologies where possible
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS, TECNALIA, UMU

Requirement ID	DK1.4	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Earth Observation Data Representation				
Description	DEMETER needs to enable the representation of current earth observation (EO) data as well as historical EO data, including for example satellite data, remote sensing imagery, soil maps, vegetation indices, such as NDVI, EVI, NDRE, NDMI. It needs to also get EO metadata, e.g., through interfaces compliant with the OGC 13-026r8[3] specification.				
Addressed by Enabler(s) / Module(s)	AIM (Semanting Mapping to OGC standard using GeoJSON, Nutrient Monitor module)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				

[3] http://www.opengis.net/doc/is/opensearch-eo/1.0

Relevant Innovation(s)	8: Unified agriculture ontology 4: Earth Observation data service
Involved stakeholders/actors	Data consumers (DSS) and processors, Data publishers, system architects, data discovery agents
Prerequisite(s)	Use Cases for EO data. EO data availability.
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Comments/Remarks	"Earth Observation Data" in this sense means remote observation of the earth, as opposed to in-situ sensors. EO data measures distribution of a phenomenon in a spatial field. There are a wide range of sensors, and a wide range of intermediate processed products, where processing may be performed on georectification, image (mosaic) aggregation, temporal averaging, cloud cover filtering, image enhancement, spectral filtering, feature detection, etc. The variability of the descriptions required and systems performing these tasks leads to a high degree of potential for interoperability challenges without a disciplined common approach. OGC provides a suite of inter-related standards for EO data encoding and provision via services; however, the semantic description aspects will require additional design work. This needs to be done in the context of a standards oriented meta-model that informs the DEMETER implementations so that consistency of approach can be achieved both within DEMETER and across other domains. OGC EO models, W3C Semantic Sensor Network and RDF-Datacube and other building blocks for this meta-model need to be adopted, adapted or mapped to, in order to maximise the long-term value of DEMETER and allow re-use of software components.

Requirement ID	DK1.5	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Representation of livestock data				
Description	DEMETER's data model needs to enable common representation of livestock data and traceability of products including: • Modelling of dairy & beef farms and data from farm robots: milk and meat production and quality, milk properties and quality (fats, proteins, somatic cells and bacterial content), economic data				

	• Modelling of data from cows' wearables: animal ID, location, temperature, pedometer data, movement. • Modelling of animals' welfare, behaviour and habits: eating habits, respiration monitoring data, rumination, activity, rectal temperature control data, feed and water consumption data, biomarkers related with animal well-being and welfare (e.g., cytokine markers) • Food traceability information of dairy products and pastries (tracking of ingredients and supply chain). • Modelling of poultry farms: animal welfare, habits, living conditions, stress levels, medical treatment, feeding patterns, feed origin. • Traceability information of poultry products (production, transport, retail) to be used in the product passport information • Modelling of apiary and hives: location of hives, apiary weight
Addressed by Enabler(s) / Module(s)	AIM (Farm Animals, Livestock Features, Poultry Feeding, Stress Recognition modules)
Relevant Pilot(s)	• Modelling of dairy & beef farms and data from farm robots: 4.1, 4.2, 4.3, 5.2 • Modelling of data from cows' wearables: 4.1, 4.2, 4.3, 5.2 • Modelling of animals' welfare, behaviour and habits: 4.1, 4.2, 4.3, 5.2 • Food traceability information of dairy products and pastries: 5.2 • Modelling of poultry farms: 4.4, 5.4 • Traceability information of poultry products: 5.4 • Modelling of apiary and hives: 5.3
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models
Relevant Innovation(s)	Innovation 8: Unified agriculture ontology Innovation 11: Data integration across the entire dairy supply chain
Involved stakeholders/actors	Solution providers, standardisation organisations
Prerequisite(s)	Data models should be based on existing ontologies
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by	ICCS, TECNALIA, ENG

Partner(s)	

Requirement ID	DK1.7	Version	1.0	Last Update Date	31/03/2021
Title	Representation of meteorological and open spatial data				
Description	DEMETER needs to enable representation of weather data (e.g., temperature, humidity, wind speed/direction, solar radiation, pressure, etc.) and open spatial data modelling. Meteorological data will be collected by interfacing with existing sensors, or new sensors that will be provided for this purpose.				
Addressed by Enabler(s) / Module(s)	AIM (Agriculture Systems and Agriculture Properties modules)				
Relevant Pilot(s)	1.4, 2.2, 3.1				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	8: Unified agriculture ontology 4: Earth Observation data service				
Involved stakeholders/actors	Solution providers, standardisation organisations				
Prerequisite(s)	Data models should be based on existing ontologies and standards				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS, TECNALIA				

Requirement ID	DK1.8	Version	1.0	Last Update Date	31/03/2021
Title	Representation of agricultural machinery data				
Description	DEMETER needs to enable common representation of agricultural machinery data such as: • engine data • fuel consumption, • emissions • exhaust gas				

	• NOx-conversion • exhaust temperatures The data are defined by Controller Area Networks Bus (CAN) protocol specifications, consequently it will be necessary to take into account that the translation of CAN-Bus Model into DEMETER Data Model, involves understanding the specific CAN Bus information (the message set for subsystem data exchange) of the supplier to the vehicle subsystem into new information according to the new DEMETER Data Model communication specifications. The new Data Model needs to represent entities types and formats, relationships among them, possible range between the values (if any).
Addressed by Enabler(s) / Module(s)	AIM (Agriculture Systems and Field Operations modules)
Relevant Pilot(s)	2.1, 2.2
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models
Relevant Innovation(s)	3: Agricultural automation and control 8: Unified agriculture ontology
Involved stakeholders/actors	Solution providers, standardisation organisations
Prerequisite(s)	Data models should be based on existing ontologies
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	TECNALIA

Requirement ID	DK1.9	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Representation of farmer's preferences and DSS recommendations to them				
Description	DEMETER needs to enable common data model able to interpret farmers' needs and preferences including: • farmers' needs related to cost optimization (e.g., linking economical aspects of wholesale and retail prices), production issues (better quality of their products, crop variety per field, optimal date for planting and harvesting), cost/benefit				

	analysis of field operations (irrigation/fertilization), optimization on irrigation/fertilization strategies, disease monitoring, yield analysis (e.g., the estimation of crop yield according to climate conditions), animal welfare tracking; • production preferences (e.g., the use of non-chemical pesticides, attention to animal welfare, transparency to the consumers); • any other relevant data input collected during farm operations (related to animal welfare, crop production, product's characteristics). DEMETER should also enable common representation of recommendations and notifications to farmers, as well as the metadata used for providing recommendations to farmers through the DSS system and analytics tools. In this way, farmers' needs and preferences will be adequately analyzed (data integration and analysis) and decision support (visualization) will be provided. Moreover, the data model to be defined will have to provide the optimization of existing DSSs, allowing them to be used by other pilots, to increase the interoperability between the Pilots through the use of a common language and syntax, to identify the entities involved, the needed relationships and attributes to define the pattern schema of the model.
Addressed by Enabler(s) / Module(s)	AIM (KPI Indicators, Stress Recognition and Poultry Feeding modules)
Relevant Pilot(s)	**1.1, 1.2, 2.2, 2.3, 2.4, 3.2, 3.3, 3.4, 4.2, 5.3**
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models
Relevant Innovation(s)	8: Unified agriculture ontology 3: Agricultural automation and control
Involved stakeholders/actors	Farmers, dairy producer, solution providers
Prerequisite(s)	None
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS

Requirement ID	DK1.11	Version	1.0	Last Update Date	31/03/2021
Title	Flexible and extensible model representation				
Description	DEMETER needs to support flexibility and extensibility in the representation of AIM through the use of a modular approach, the reuse or alignment with thesauri/classifications available as linked data, the use of property graphs and semantics, the use of appropriate data interchange models (e.g., RDF), knowledge representation languages (e.g., SKOS, RDFS, OWL) and rule languages (e.g., SWRL or OWL-RL), which would enable the semantic querying of data.				
Addressed by Enabler(s) / Module(s)	AIM (NGSI-LD layered approach)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	8: Unified agriculture ontology				
Involved stakeholders/actors	Ontology engineers, semantic technologies experts				
Prerequisite(s)	Competency questions, pilots' requirements				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS, PSNC, ENG, FhG.FIT, TECNALIA, OGCE				

Requirement ID	DK1.13	Version	1.0	Last Update Date	31/03/2021
Title	Representation of data quality metrics				
Description	DEMETER needs to include quality metrics in its data model. This data will be used for evaluating the accuracy, precision, granularity, completeness, consistency, timeliness, validity, uniqueness (where applicable) of the agrifood data and will be used by the data fusion and analytics tools that Demeter provides.				
Addressed by	AIM (KPI Indicators)				

Enabler(s) / Module(s)	
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models
Relevant Innovation(s)	8. Unified agriculture ontology
Involved stakeholders/actors	Solution providers
Prerequisite(s)	DK1.1, Data/information for implementing metrics
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS

Requirement ID	DK1.14	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Provide a basis for data exchange across stakeholders				
Description	DEMETER needs to enable data exchange across authorised stakeholders. To facilitate this, it needs to include data regarding the supply and usage of agri-data and any other type of data that is stored in the DEMETER unified ontology including any economic transactions regarding the usage of such data.				
Addressed by Enabler(s) / Module(s)	AIM (Semantic mappings to well-known ontologies)				
Relevant Pilot(s)	All				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models Objective 2: Build data knowledge exchange mechanisms				
Relevant Innovation(s)	8. Unified agriculture ontology 9. Secure Agricultural data sharing services				
Involved stakeholders/actors	Authorised stakeholders				

Prerequisite(s)	Gathered data
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS

Requirement ID	DK1.15	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Data Models enabling analysis of large heterogeneous data				
Description	DEMETER needs to provide data models that will enable the analysis and processing of large amount of heterogeneous data, including their storage and transfer. DEMETER should take advantage of numerous sources, like wireless sensor networks and imagery to store, fuse and process all the data that are required by Demeter applications; see DK1.3-9 for details.				
Addressed by Enabler(s) / Module(s)	AIM				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	8. Unified agriculture ontology				
Involved stakeholders/actors	Pilot leaders, domain experts, ontology engineers				
Prerequisite(s)	Competency questions, pilots' data requirements				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	ICCS, PSNC, ENG, FhG.FIT, TECNALIA, OGCE				

Requirement ID	DK1.16	**Version**	1.0	**Last Update**	31/03/2021

				Date	
Title	Model for Data Brokerage Services				
Description	DEMETER needs to provide a common information model for Data Brokerage Services. It is necessary to have enough data in order to describe the offered resources and Demeter-enabled entities, their capabilities as well as the policies of those that offer them including pricing policies. In addition, once such entities are contracted it is necessary to keep transaction and (potentially) whole supply chain information in the model. These could even include the representation of (fragmented) supply chain stakeholder information.				
Addressed by Enabler(s) / Module(s)	WP3 service metadata model				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models Objective 2: Build data knowledge exchange mechanisms				
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified agriculture ontology				
Involved stakeholders/actors	Consumers, producers				
Prerequisite(s)	-				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	LESP				

Requirement ID	DK1.17	**Version**	1.0	**Last Update Date**	31/03/2021
Title	General Model for Interoperability				
Description	DEMETER needs to provide a general model for data interoperability, which should be flexible and extensible for all use cases. More specifically: 1. It will be composed of discrete modules addressing specific “competency questions”, following best practices in ontology engineering- allowing these to				

	be adopted standards or tightly managed development efforts with clear testability. 2. It needs to handle interoperability for different implementation aspects. 3. Meta-models, domain models, profiles and vocabularies need to be handled individually using appropriate specialised modelling mechanisms.
Addressed by Enabler(s) / Module(s)	AIM
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models Objective 2: Knowledge Exchange Mechanisms
Relevant Innovation(s)	1. Agriculture Interoperability Space 8. Unified Agriculture Ontology
Involved stakeholders/actors	Data modelers, component system designers, system architects, standardization organizations
Prerequisite(s)	-
Type	Methodology
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	OGCE, FhG.FIT
Comments/Remarks	DEMETER will require a complex set of interoperability agreements covering many different aspects of data exchange and semantic description. Common approaches tend to be one-dimensional – i.e., limited to a single model of interoperability – such as service interfaces, data schemas, openness and accessibility, etc. To achieve integration of component systems across the DEMETER project scope, it will be necessary to achieve interoperability across a range of such concerns. At this stage, it is not easy to identify a comprehensive model of interoperability that can be adopted, however by capturing each aspects where stakeholders need information about some aspect of a data exchange, and the role of various supporting infrastructures and components it will be possible to develop and exemplify a minimum necessary and sufficient interoperability architecture. The diversity of pilots and functional requirements in DEMETER demands and provides an opportunity to test innovation in this space.

Requirement ID	DK1.18	Version	1.0	Last Update Date	31/03/2021
Title	Simplified Profiles of Data Model				
Description	DEMETER needs to provide simple profiles of the general model suitable for individual pilot cases; these profiles will define “schemas” - or views, while the general model will define semantics - what objects can be identified and reused in different views.				
Addressed by Enabler(s) / Module(s)	AIM				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	8. Unified agriculture ontology				
Involved stakeholders/actors	Data modelers, implementers of systems, data integrators, data discovery agents, semantic technology experts, pilots’ data leads				
Prerequisite(s)	Comprehensive data model, profiling mechanism, identification of stakeholders’ requirements for simplified application specific subset of model.				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	OGCE				
Comments/Remarks	The DEMETER model, as suggested by the full range of requirements, will be a very large and complex, highly modular model using some powerful but very general standards. When implementing or using data for a specific component (via some API or data exchange) only a small subset of this model will be involved. The concept of “profile” is used to narrow general models down to specific simplified cases. The formalism of the profile descriptions allows simple views to be discovered and integrated safely into the common, complete, model as required. Profiles are akin to mapping (and may include mapping specifications) but they also allow for the simpler and less semantically ambiguous case of just selecting a relevant subset of the common model for use in a particular Use Case.				

Requirement ID	DK1.19	Version	1.0	Last Update Date	31/03/2021
Title	Semantic model that supports scalability and support of legacy systems				
Description	DEMETER needs to implement semantic interoperability in a scalable and sustainable way, e.g., by maintaining a dependency graph at the module level within each implementation rather than creating a temporary (project scoped) aggregated knowledge graph with no transparency of scope or provenance. It should support semantic interoperability for data originating from existing systems involved in the pilots (legacy systems). It should publish all domain-specific semantic interoperability resources in a canonical standards-based and interoperable fashion appropriate to the type of resource (e.g., vocabulary, schema, object model, profile, datatype, etc.)				
Addressed by Enabler(s) / Module(s)	AIM				
Relevant Pilot(s)	**All**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	8: Unified agriculture ontology				
Involved stakeholders/actors	Data publishers, system architects, infrastructure providers, standards organizations.				
Prerequisite(s)	Semantics publishing activities.				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	OGCE, ICCS				
Comments/Remarks	The complexity of the DEMETER project scope, and its interactions with related domains of knowledge, means that a significant number of components will be required to implement the range of pilot projects. Many of these components will be information resources, such as ontologies, vocabularies, mappings and registers which will be used to drive data integration and processing functions. In order to keep these simple, the number will grow (there is a trade-off between a few very				

	complex objects and a large number of simple objects). Experience has shown that small sets of complex objects are hard to maintain, but large sets of simpler objects require tools and infrastructure to manage dependencies. In the Java world, the Maven infrastructure manages dependencies across the typically hundreds of small libraries – but developers, once they embrace this, are freed from the curse of continually changing critical libraries and extreme complexity and risk in assessing what impacts changes may have. DEMETER should "plan to succeed" by assuming a sophisticated dependency management approach as a core architectural requirement.

Requirement ID	DK1.20	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Governance Arrangements				
Description	DEMETER needs to specify governance arrangements for each component, determining who, when and how updates to the included components should be handled. This includes pragmatic project-scope governance of temporary resources, as well as requirements for governance of project resources that would ensure future interoperability.				
Addressed by Enabler(s) / Module(s)	Issue Tracker, Guidelines				
Relevant Pilot(s)	**All**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models				
Relevant Innovation(s)	9: Secure Agricultural data sharing services 8: Unified agriculture ontology				
Involved stakeholders/actors	Data publishers, system architects, infrastructure providers, standards organizations.				
Prerequisite(s)	Project architecture and persistence requirements.				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	OGCE				

Comments/Remarks	To have a lasting impact and application beyond the initial pilots there needs to be a means to share semantic resources amongst a community of practice in a sustainable fashion. Resources need to meet FAIR principles (Findable, Accessible, Interoperable and Reusable). At the heart of "Reusable" is the issue of risk and trust – in order to reuse a resource a stakeholder needs to understand both the technical usefulness of the resource as well as how it may be "Accessible" in future. One aspect of this is "governance" – as exemplified by ISO 19135 (Procedures for Item Registration). Transparency of governance includes the policies and mechanisms by which resources may be created, reviewed and updated, and should also include statements around the persistence of identifiers and services.

Requirement ID	DK1.21	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Abstract model for integrating sensors, processing and decision support systems.				
Description	DEMETER needs to have an abstract model for the general process of linking sensor data through processing chains into decision support systems, including how intermediate data products relate to sources and outputs. This can be based on an existing general model, or, if necessary, to create something new, to be pushed as an OGC and/or W3C general model spec.				
Addressed by Enabler(s) / Module(s)	AIM (Semantic mapping and profiling of SOSA/SSN standard)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models Objective 2: Knowledge Exchange Mechanisms				
Relevant Innovation(s)	5. Farm enabler dashboards 7. Cost- and power-effective IoT data acquisition 8. Unified agriculture ontology 14. Smart fruit pesticides management				
Involved stakeholders/actors	Data integrators, data discovery agents.				
Prerequisite(s)	General interoperability model (DK1-17), Interoperability profiles for abstract components (sensors, processes etc.)				
Type	Functional				

Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	OGCE
Comments/Remarks	Many DEMETER pilots explicitly or implicitly involve aggregation of data into a decision support system (DSS). In general, it is assumed that DSS will have the ability to reuse available data, and such data may be used by many systems, including DSS. Usually, if not always, data from direct observation (including sensors) is processed in a pipeline to be delivered to the DSS in a form relevant to the decision criteria. Aggregation, interpolation, and signal (pattern) detection in time and space is usually performed to simplify rich observation data into summaries of state and trends of conditions of interest. In order to have reuse and interoperability of data and processing systems, the role of each in relation to both the DEMETER information model and the functional processing involved needs to be described. This will be relatively complex – and, if this is done on an ad-hoc basis, the effective complexity (ability to identify reusability) will grow exponentially with the number of examples. If simplified profiles of the data model are identified for a set of abstract processes (i.e., a profile for measurement of some environmental factor at farm scale at regular intervals over a year) - then the level of complexity grows more slowly, as much of the descriptive burden is handled by reusable patterns for common processes, and it will become possible to compare and reason over data descriptions based on similarity as well as specifics. Crudely, it should be possible to ask a data catalog what data sources are compatible with a given DSS input requirement. This will require a model of both such DSS data requirements and available processing steps that can be used to generate the data product from observational data. At the very least, capturing the DEMETER pilot scope will inform the development of a generalized approach for future interoperability standards.

7.2 Semantic mapping of AIM to dominant/standardised agrifood solutions

Requirement ID	DK2.1	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Service wrappers and translators				
Description	DEMETER needs to develop service wrappers and translators, also known as DEMETER providers and consumers, which will enable the different tools/platforms in a (regional/national) AKIS to expose and consume data in interoperable forms.				

Addressed by Enabler(s) / Module(s)	Wrappers and translators from and to AIM (JSON, CSV, XML)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.2
Relevant Objective(s)	O2: Build knowledge exchange mechanisms O6: Demonstrate the impact of digital innovations across a variety of sectors and at European level
Relevant Innovation(s)	1. Agriculture Interoperability Space 11. Data integration across the entire dairy supply chain 15. Open AKIS for irrigated crops 17. Water Management Model and Coordination Broker
Involved stakeholders/actors	Solution providers, semantic technologies experts
Prerequisite(s)	AIM
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	PSNC, ICCS, OGCE, TECNALIA
Comments/Remarks	Such components include for instance, (a) components enabling to harness satellite data for applications in farm telemetry, with particular interest in crop monitoring and predictions, (b) components for crop monitoring and real-time analytics using real-time streaming data from wireless sensor networks, and (c) components with the capability to trigger alarms, notifications and/or recommendations in order to improve farm operations and productivity. The DEMETER provider-consumer services, deployed on the various components, translate and exchange data based on the AIM common data format with the utilization of lightweight data wrappers/translators. Hence, in order to develop these wrappers/translators each of the AKIS components should provide the specifications of the utilized data model-semantics and/or it should parse the returned content in AIM format. The translators will then implement mapping rules between the components' underlying data models and AIM to transform the data from the component to/from AIM ontology. This may also include syntactic and data conversion rules (e.g., mapping to common datum, timezones, etc).

Requirement ID	DK2.2	Version	1.0	Last Update Date	31/03/2021
Title	Mapping AIM with standard models				
Description	DEMETER should implement (semantic) mappings from standard and/or widely used ontologies/vocabularies with the AIM, enabling the semantic integration of data represented using any of these models. As part of the semantic mapping, DEMETER will need to identify logical connections between classes, properties, and objects across ontologies. The mappings will deal with cases in which, e.g., a class in one ontology is the intersection (or union) of two classes in another, or the complement of another class, or a simple object needs to be mapped to a complex class in another ontology, etc.				
Addressed by Enabler(s) / Module(s)	AIM (Semantic mapping to FIWARE, FOODIE, SAREF4AGRI, ADAPT, INSPIRE, EPPO, AGROVOC, etc.)				
Relevant Pilot(s)	**ALL**				
Relevant Task(s)	T2.1				
Relevant Objective(s)	O1: Analyse, adopt, enhance existing (and if necessary, introduce new) information models O2: Build knowledge exchange mechanisms				
Relevant Innovation(s)	8. Unified agriculture ontology				
Involved stakeholders/actors	Domain experts, Semantic technologies experts, data consumers				
Prerequisite(s)	AIM				
Type	Functional				
Status	Addressed				
Priority Level	Mandatory				
Identified by Partner(s)	PSNC, ICCS, OGCE				
Comments/Remarks	Data consumers benefit by being able to refer to standard models and terms in order to understand data content				

Requirement ID	DK2.3	Version	1.0	Last Update Date	31/03/2021
Title	Semantic Interoperability				

Description	Support semantic interoperability, encompassing semantic integration (DK3). This will be realized through the implementation of the various DEMETER provider and consumer services (DK3.1), new ontologies and the mappings with existing ontologies/vocabularies (DK3.2), as well as the other mechanisms developed to facilitate data integration (see DK3).
Addressed by Enabler(s) / Module(s)	AIM (Semantic mappings and alignments)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1
Relevant Objective(s)	Objective 1: Analyse, adopt, enhance information models.
Relevant Innovation(s)	8.Unified agriculture ontology
Involved stakeholders/actors	Consumers, Producers
Prerequisite(s)	Existing ontologies/vocabularies, DK3.1, DK3.2, publishing mechanisms and standards for components required to publish in full.
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS
Comments/Remarks	This requirement implies full support for all semantic information through a complete data publishing, access, integration and use lifecycle (discovery, publishing, analysis, notification, etc. of the results of the integration are related but distinct requirements). This requirement focuses on whether sufficient information is available to support integration and whether that information is accessible and interoperable.

Requirement ID	DK2.4	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Mapping best practices				
Description	Follow best practices and approaches for generating the mapping between AIM and existing ontologies/vocabularies, including: • Transformation of existing ontologies into common format, e.g., OWL, use of semantic rules or annotations/punning.				

	• Reuse of AIM terms, and only extend it if necessary. In the latter case, reuse existing terms whenever possible, and only otherwise create new terms/extensions. • Use of appropriate mapping constructs/axioms, such as owl:equivalentClass and owl:equivalentProperty with OWL classes/properties; skos:closeMatch, skos:exactMatch, skos:broadMatch, skos:narrowMatch, and skos:relatedMatch with SKOS concepts; owl:sameAs for individuals, etc. • Treating of the mappings as “first class” components of a modular knowledge graph, making them available in line with FAIR principles, and governing them appropriately and transparently. • Consider mappings across different levels of specification granularity as well of abstractions using the appropriate mechanisms in a standardised way, e.g., mappings from meta-models to models (OWL subclassing); mappings between concepts at the same level of abstraction; mappings between controlled vocabulary terms; mappings between measurements and classifications (e.g., threshold values for "good" etc.); soft- vs. hard-typing mappings with classes with a sub-type property vs. specific sub-classes
Addressed by Enabler(s) / Module(s)	AIM (Semantic mappings and alignments, AIM wrappers)
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1
Relevant Objective(s)	O1: Analyse, adopt, enhance existing (and if necessary, introduce new) information models
Relevant Innovation(s)	8. Unified agriculture ontology
Involved stakeholders/actors	Domain experts, Semantic technologies experts
Prerequisite(s)	AIM
Type	Functional
Status	Partially addressed
Priority Level	Mandatory
Identified by Partner(s)	ICCS, PSNC, OGCE, Tecnalia

Requirement ID	DK2.5	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Tools for generating ontology mappings (semi-) automatically				

Description	Identify and select, if possible, suitable tools for the (semi-) automatic mapping of ontologies/vocabularies. Some example tools to be analysed include the Alignment API, PARIS and Map-On
Addressed by Enabler(s) / Module(s)	VocBench
Relevant Pilot(s)	• Information model for water management: 1.1, 1.2, 1.3, 1.4, 3.1, 3.2 • Information model of crops, pests, treatment and fertilisation data: 1.3, 1.4, 2.2, 3.1, 3.2, 3.3, 5.1, 5.3 • Information model of soil data: 1.4, 3.2 • Information model for weather data: 1.4, 2.2, 3.1 • Information model of Vehicle data and emissions: 2.1 • Information model for farms and animals: 4.1, 4.2, 4.3, 4.4, 5.2, 5.3, 5.4 • Information model of farm economic data: 2.4 • Information model of status and field data: 1.3, 3.1, 3.4, 5.2 • Information model for the traceability of crops, dairy products, poultry products: 5.1, 5.2, 5.4
Relevant Task(s)	T2.1
Relevant Objective(s)	O1: Analyse, adopt, enhance existing (and if necessary, introduce new) information models
Relevant Innovation(s)	8: Unified agriculture ontology
Involved stakeholders/actors	Solution providers, Domain experts, Semantic technologies experts
Prerequisite(s)	Data models should be based on existing ontologies
Type	Functional
Status	Partially addressed
Priority Level	Desirable
Identified by Partner(s)	PSNC, OGCE, ICCS, TECNALIA

Requirement ID	DK2.6	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Identify tools to validate mappings				

Description	In order to facilitate the mapping between the Demeter AIM and existing ontologies, it is necessary to identify and select, if possible, suitable tools to validate the generated mappings. This is necessary because some of the mappings may be quite complex. For example, when a specific schema is mapped to a more general schema, then some schema elements may be replaced by use of a qualifying term in corresponding more abstract elements. In such cases, we need to validate the coverage of the mappings as well as the result of exercising a mapping against the target model. It would also be desirable to define a validation process and a simple reference implementation that can define test procedures to be integrated into traditional development tooling.
Addressed by Enabler(s) / Module(s)	pySHACL, Apache Jena SHACL, Astrea Web Service
Relevant Pilot(s)	**ALL**
Relevant Task(s)	T2.1
Relevant Objective(s)	O1: Analyse, adopt, enhance existing information models O2: Build knowledge exchange mechanisms
Relevant Innovation(s)	8. Unified agriculture ontology
Involved stakeholders/actors	Domain experts, Semantic technologies experts
Prerequisite(s)	Standardised approaches to publishing schema, terms, mappings between schema, mappings between terms and binding of term ranges to schema (profiles of schema)
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	PSNC, ICCS, OGCE
Comments/Remarks	• Different tools may be required for different aspects. Mappings may be trivial - can be limited to schema mappings or term mappings - but many may be more complex - for example when a specific schema is mapped to a more general schema (the usual case) then some schema elements may be replaced by use of a qualifying term in corresponding more abstract elements:

	<pig>123</pig> => <animal><type>swine</type><id>123</id></animal> Tools such as VocBench may support effective means of validating term mappings, and schema mapping tools may be available that can be adapted. Tools should be evaluated on multiple criteria: 1) which parts of mappings they can validate 2) effectiveness at validation 3) ease of integration into testing mechanisms 4) accessibility to relevant stakeholders 5) flexibility 6) overhead of familiarisation with tool-specific UI and data management paradigms In general, however, a validation process needs to be defined and a simple reference implementation that can define test procedures needs to be integrated into traditional development tooling - i.e., wrapped up as a test case with test case data samples and executed using readily available orchestration tools. A ubiquitous language like python and tools like pySHACL can be used to validate input and output shapes using available standard constraints languages. A regression testing using some form of continuous integration will be required to ensure that evolving quality and scope of mappings for more complex cases continue to work reliably for the simpler cases that will probably be validated and deployed first.

Requirement ID	DK2.7	**Version**	1.0	**Last Update Date**	31/03/2021
Title	Select relevant existing ontologies to align with AIM				

Description	Identify and select relevant standards and/or widely used ontologies/vocabularies to align with the AIM and identify the key terms in each of them that would need to be aligned. Some examples, classified by the type of data, include:
	• Water management:
	  ○ Saref4agri → s4agri:WateringSystem
	  ○ INSPIRE → WaterManagement, irrigation
	• Crops and pests:
	  ○ FOODIE (INSPIRE based) → cropType
	  ○ FIWARE → AgriCrop, AgriPest
	  ○ rmAgro
	  ○ drmCrop				

	◦ AGROVOC → crops, plant products • Soil data and other sensor measurements: ◦ Saref4agri → s4agri:SoilMoisture, s4agri:SoilTemperature ◦ AFarCloud → observations (IoT devices) ◦ Soilphysics → soilProperty ◦ SOSA/SSN → sosa:Observation • Weather data: ◦ Agrifood Data Models → Weather Observed, Weather Forecast, Weather Alert ◦ Saref4agri → s4agri:AmbientHumidity, s4agri:AirTemperature • Vehicle data: ◦ FOODIE → Transport data model ◦ AFarcloud → hierarchy of robotic vehicles (UGVs, AUvs) • Farm data: ◦ FOODIE → Holding, Site, Plot, ManagementZone, ProductionType, CropSpecies, etc. ◦ AgriFarm ◦ Saref4agri → s4agri:Farm • Field data ◦ FOODIE → cropSpecies ◦ FIWARE → AgriCrop ◦ Saref4agri → s4agri:Crop, s4agri:PlantGrowthStage; • Animals and dairy farms ◦ Agrifood → Animal ◦ Saref4agri → s4agri:Animal, s4agri:MilkingSensor, s4agri:ActivitySensor ◦ INSPIRE → Animals and animals health ◦ AFarCloud → Dairy farms • Treatments ◦ FOODIE → Treatment, TreatmentPlan
Addressed by Enabler(s) / Module(s)	AIM (Semantic mappings and alignments, AIM wrappers)
Relevant Pilot(s)	• Information model for water management: 1.1, 1.2, 1.3, 1.4, 3.1, 3.2 • Information model of crops, pests, treatment and fertilisation data: 1.3, 1.4, 2.2, 3.1, 3.2, 3.3, 5.1, 5.3 • Information model of soil data: 1.4, 3.2

	• Information model for weather data: 1.4, 2.2, 3.1 • Information model of Vehicle data and emissions: 2.1 • Information model for farms and animals: 4.1, 4.2, 4.3, 4.4, 5.2, 5.3, 5.4 • Information model of farm economic data: 2.4 • Information model of status and field data: 1.3, 3.1, 3.4, 5.2 • Information model for the traceability of crops, dairy products, poultry products: 5.1, 5.2, 5.4
Relevant Task(s)	**T2.1**
Relevant Objective(s)	Objective 1: Information Modelling
Relevant Innovation(s)	3. Agricultural automation and control 8: Unified agriculture ontology 11: Data integration across the entire dairy supply chain 17: Water Management Model and Coordination Broker
Involved stakeholders/actors	Solution providers
Prerequisite(s)	Existing relevant ontologies
Type	Functional
Status	Addressed
Priority Level	Mandatory
Identified by Partner(s)	PSNC, OGCE, ICCS, TECNALIA

8 Revised design of the Agricultural Information Model (AIM)

In line with best practices and recommendations, the specification of DEMETER AIM follows a modular approach in a layered architecture, enabling among others:

1. straight-forward interoperability with existing models by reusing available (well-scoped) models in the modules, instead of defining new terms, whenever possible,
2. easy mapping/alignment with other models, by module instead of the entire model,
3. easy extension of the domain/areas covered in AIM with additional modules,
4. easy extension of the domain model, by modifying only specific modules,
5. easy mapping to top-level/cross-domain ontologies.

An overview of the second release of the Agricultural Information Model (AIM) is provided in Figure 2.

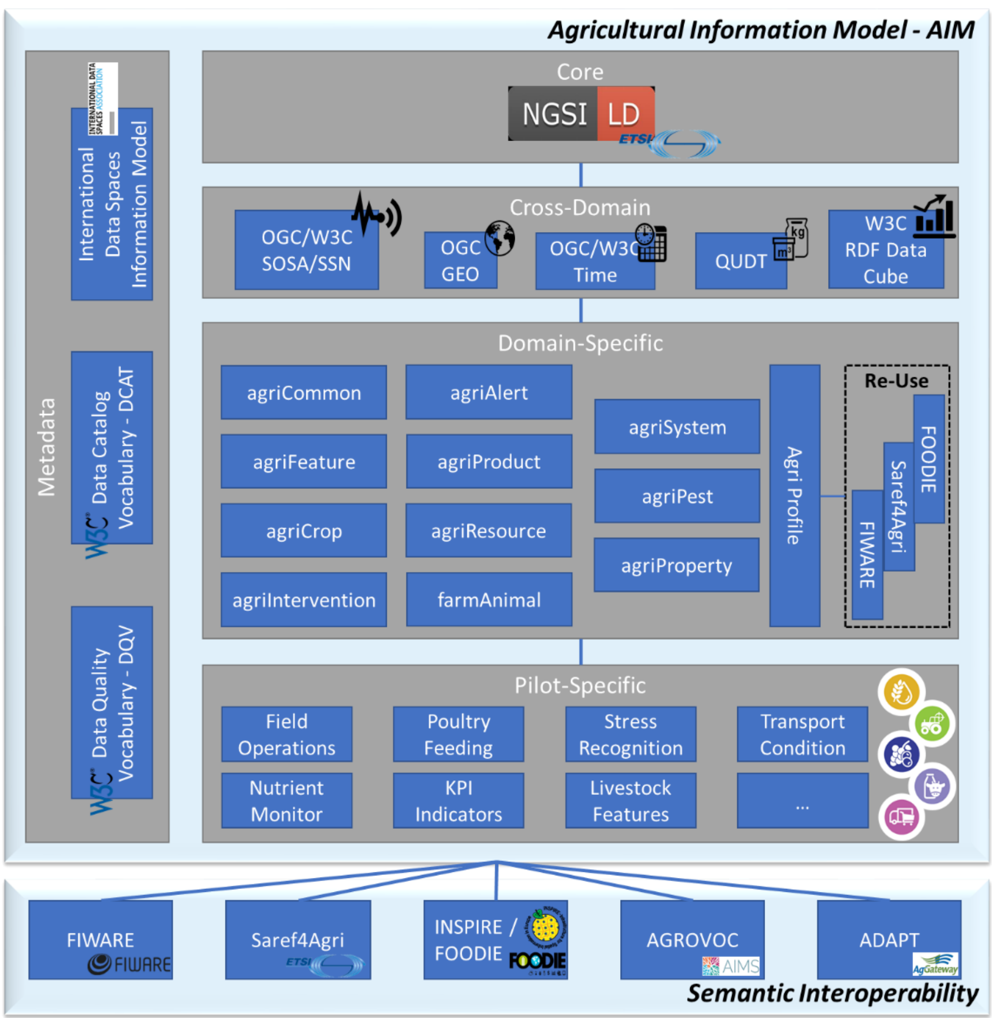

Figure 2. Overview of the layers of the second release of the DEMETER Agricultural Information Model (AIM)

Every AIM layer in the initial release underwent several changes aiming for increased alignment with well-known ontologies, optimal exploitation of the modularity and portability of the AIM design and easier adaptation of external resources. It should also be highlighted that AIM has been updated with an additional layer, the pilot-specific layer, which is situated below the domain-specific layer. Section 8.4 presents this new layer and describes the motivation behind this necessary revision, while the rest of the subsections present the revised versions of the other four AIM layers, the initial version of which is elaborated upon in Deliverable 2.1.

8.1 Core meta-model

A meta-model, as its names implies, is a model of a model. Meta-models are typically used for different purposes. For instance, they can be used for the specification of modelling language constructs in a standardized, platform independent manner [HaPa09], to specify and restrict a domain in a data model and systems specification [IvVo11], or to provide an explicit model of the constructs and rules needed to build specific models within a domain of interest [Wel]. In fact, as noted in [Wel], meta-models can be viewed from three different perspectives: i) as a set of building blocks and rules used to build models; ii) as a model of a domain of interest; iii) as an instance of another model. In the context of the DEMETER meta-model, we are considering it as the first perspective.

As discussed and described in detail in the first release of the model, in deliverable D2.1, AIM follows the NGSI-LD meta-modeling approach [NGS1]. For more details on this, please refer to Annex C, where the analysis of the core meta-model of AIM is provided, along with the specifics of NGSI-LD. Similar information can be found also in Deliverable 2.1, though the code snippets are updated and aligned with the current version of AIM.

DEMETER AIM initially followed the same 3-layer architecture of NGSI-LD, including a property graph meta-model layer (grounded in RDF/RDFS), a cross-domain ontologies layer, and the domain/application ontologies. This has been extended with an additional layer comprising DEMETER's pilot specific extensions, as described in previous section. Though, as opposed to NGSI-LD, DEMETER AIM implements the cross-domain and domain/application layers by reusing existing standards and/or well-known ontologies/vocabularies as much as possible from the outset, thereby implementing semantic referencing, as it is described in Section 8.2 and 8.3.

The AIM core meta-model underwent no major changes since its description in D2.1. There have always been changes related to the avoidance of conflicts with other layers. More specifically, the json-ld contect was updated using the prefix "meta" (e.g., meta.Property) as diacritical mark. For the same reason, the mapping of "id" and "type" to json-ld keys was removed, as they are extensively used in other layers.The DEMETER core meta-model is published as a Turtle (TTL) file at:

https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/coremetamodel

8.2 Cross-Domain ontology

The general motivation behind the AIM cross-domain layer is, to:

- Capture concepts and terms that are generic and applicable to various domains.
- Avoid conflicting or redundant definitions of the same concept in different domain specific models.
- Provide a basis for interoperability with other information systems and tooling.

Based on these motivations, the Cross-Domain model has undergone changes since Deliverable 2.1. Conceptual changes in the domain-specific layer, primarily based on pilot needs, needed to be propagated to the cross-domain layer in order to capture terms applicable to multiple domains. Along with the conceptual updates, some minor technical improvements to the schema were conducted, which are discussed in Chapter 10.2.

8.2.1 Conceptual changes in second release of the AIM Cross-Domain ontology

Conceptual changes to the Cross-Domain Ontology can be summarized as follows.

Added Concepts:

Concepts were either added based on pilot needs or to improve the alignment with the domain layer, as indicated in the following table.

Table 1. Added concepts in the Cross-Domain layer of the second AIM release

Ontology	Concept	Reason
SOSA – Sensor, Observation, Sampling, Actuator Ontology (https://www.w3.org/ns/sosa/)	sosa:hasResult	Pilot need
	sosa:observes	Pilot need
	sosa:phenomenonTime	Pilot need
	sosa:usedProcedure	Pilot need
	sosa:hasUltimateFeatureOfInterest	Pilot need
	sosa:hasMember	Pilot need
	sosa:Result	Pilot need
	sosa:Sample	Pilot need
	sosa:ObservationCollection	Pilot need
	sosa:Sensor	Pilot need
	sosa:Sampling	Pilot need
	sosa:Actuation	Pilot need
	sosa:Procedure	Pilot need
	sosa:hasSimpleResult	Pilot need
	sosa:resultTime	Pilot need

SSN – Semantic Sensor Network (https://www.w3.org/TR/vocab-ssn/)	ssn:hasInput	Pilot need
	ssn:hasOutput	Pilot need
	ssn:implementedBy	Pilot need
QUDT - Units of Measure, Quantity Kinds, Dimensions and Data Types (http://www.qudt.org/pages/HomePage.html)	qudt:unit	Pilot need
	qudt:quantityValue	Pilot need
	qudt:hasQuantityKind	Pilot need
	qudt:numericValue	Pilot need
	qudt:Quantity	Pilot need
	qudt:QuantityKind	Pilot need
	qudt:QuantityValue	Pilot need
	qudt:Unit	Pilot need
Basic Geo (WGS84 lat/long) Vocabulary (https://www.w3.org/2003/01/geo/)	wgs84_pos:lat	Pilot need
	wgs84_pos:long	Pilot need
	wgs84_pos:alt	Pilot need
OGC Time Ontology in OWL (https://www.w3.org/TR/owl-time/)	time:TemporalEntity	General need for time ontology
	time:TemporalDuration	
	time:TemporalUnit	
	time:Interval	
	time:Instant	
	time:Duration	
	time:hasBeginning	
	time:hasEnd	
	time:hasTime	
	time:inXSDDate	
	time:inXSDDateTimeStamp	
	time:numericDuration	
	time:unitType	
	time:unitDay	
	time:unitHour	
	time:unitMinute	
	time:unitMonth	
	time:unitSecond	
	time:unitWeek	
	time:unitYear	
ISO 19156:2011 Geographic information —	iso19156_GFI:GFI_Feature	Replaced

Observations and measurements (https://www.iso.org/standard/32574.html)	iso19156_GFI:GFI_DomainFeature iso19156_SF:SF_SamplingFeature iso19156_OB:OM_Observation iso19156_SF:SF_SamplingFeature.sampledFeature	old ISO19150 elements from FOODIE, by more current ISO19156 elements
ISO 19103:2015 Geographic information — Conceptual schema language (https://www.iso.org/standard/56734.html)	iso19103:Measure	Pilot need
Ontology for Quantity Kinds and Units: units and quantities definitions (https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu-rec20.html)	dim:HeatScheme dim:ElectricityAndMagnetismScheme dim:PhysicalChemistryAndMolecularPhysicsScheme dim:SolidStatePhysicsScheme dim:LightAndRelatedElectromagneticRadiationsScheme quantity:criticalBuildUpSpeed quantity:criticalTorsionalSpeed quantity:criticalWhirlingSpeed quantity:pseudovectorProperty quantity:rotationalSpeed quantity:synchronousPullOutTorque quantity:synchronousSpeed quantity:torque	Alignment with domain layer

Removed Concepts:

Table 2. Removed concepts in the Cross-Domain layer of the second AIM release

Concept	Reason
xsd:date	Replaced with OGC OWL Time
geo:hasSerialization	Not needed
iso19109:AnyFeature	Not needed

iso19150-2:FeatureType	Replaced old ISO19150 elements from FOODIE, by more current ISO19156 elements

8.2.2 Overview of the second release of the AIM Cross-Domain ontology

Figure 3 presents an overview on the Cross-Domain model, indicating which parts of the model changed/were added in course of the revision. The model is centred around the NGSI-LD Entity concept, which provides a superclass for all concepts contains in the model. This provides the link to AIM core, which defines the NGSI-LD Entity as well as the related concepts NGSI-LD Relationship and NGSI-LD Property. The Figure denotes that that three ontologies (SSN-ext, W3C OWL Time, and ISO 19156 Geo) were added to the model, while two existing ontologies were changed (or extended). Additional terms from SSN/SOSA and QUDT were adopted to meet pilot needs and to unify new cross-domain terms arisen in the domain-specific layer.

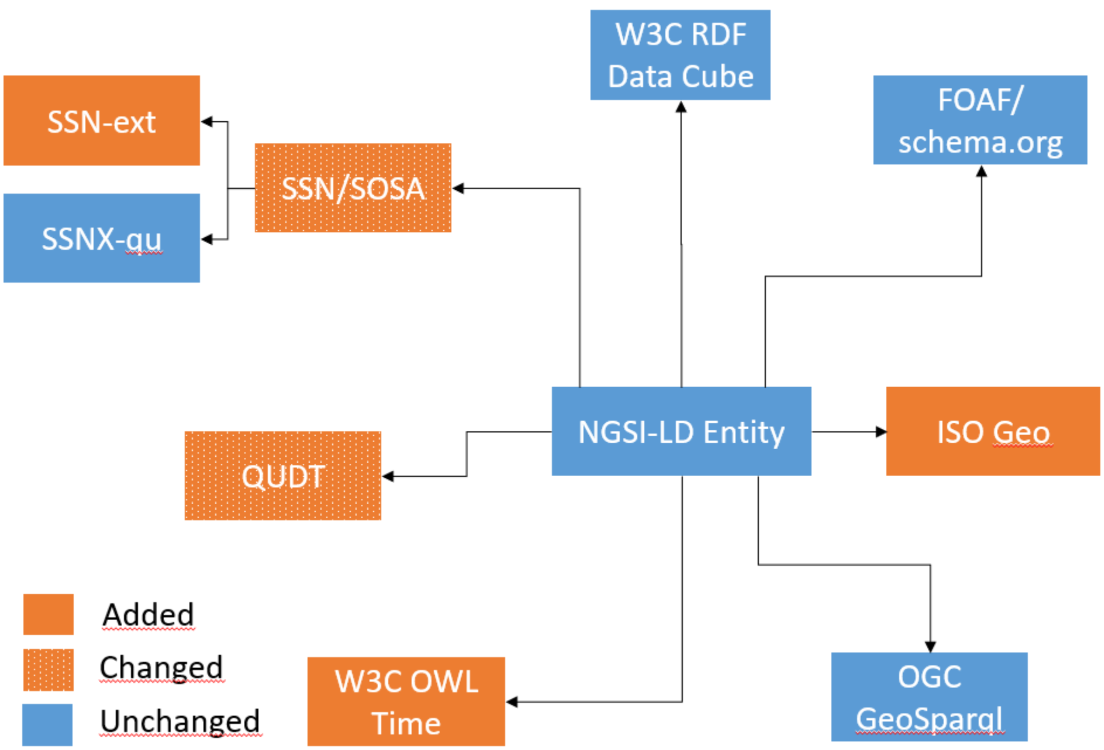

Figure 3. Overview of the second release of the AIM Cross-Domain ontology

In summary, concepts from the respective ontologies serve the following purpose:

- **W3C OWL Time** contains concepts of temporal properties and time values.
- **OGC GeoSPARQL** and associated definitions describe geographical and geometrical properties.
- **The W3C/OGC** recommendation **SOSA/SSN** contains concepts regarding sensor and actuator data, including observations, observation collections, observed properties, systems and platforms.
- **QUDT** contains concepts regarding units of measurement, and concepts to represent quantities and quanity kinds.
- Concepts from the **W3C RDF Data Cube** vocabulary are being used to represent statistical data, including datasets, data structures, slices, measure properties, dimension properties, etc.
- Alignment with **ISO geographic technology standards (ISO Geo)**, includes features (domain and sampling features), and observations.
- Basic terms from standard or widely used vocabularies like **FOAF, schema.org**.
- Alignment with core meta-model layer (**NGSI-LD**)

8.3 Domain-Specific ontologies

The structure of the domain-specific ontology layer is presented in the figure below. Even though there have been no changes concerning the high-level design of the layer, there are some changes that have been made over four of the components of the domain-specific ontology layer, namely: the Agriculture Commons Ontology, the Agriculture Features Ontology, the Agriculture Properties Ontology and the Agriculture System Ontology. The description of the changes over these ontologies with regards to their initial release is provided herewith. The ontologies that remain unchanged are described in detail in Deliverable 2.1 and the complete layer is presented in detail in Annex D.

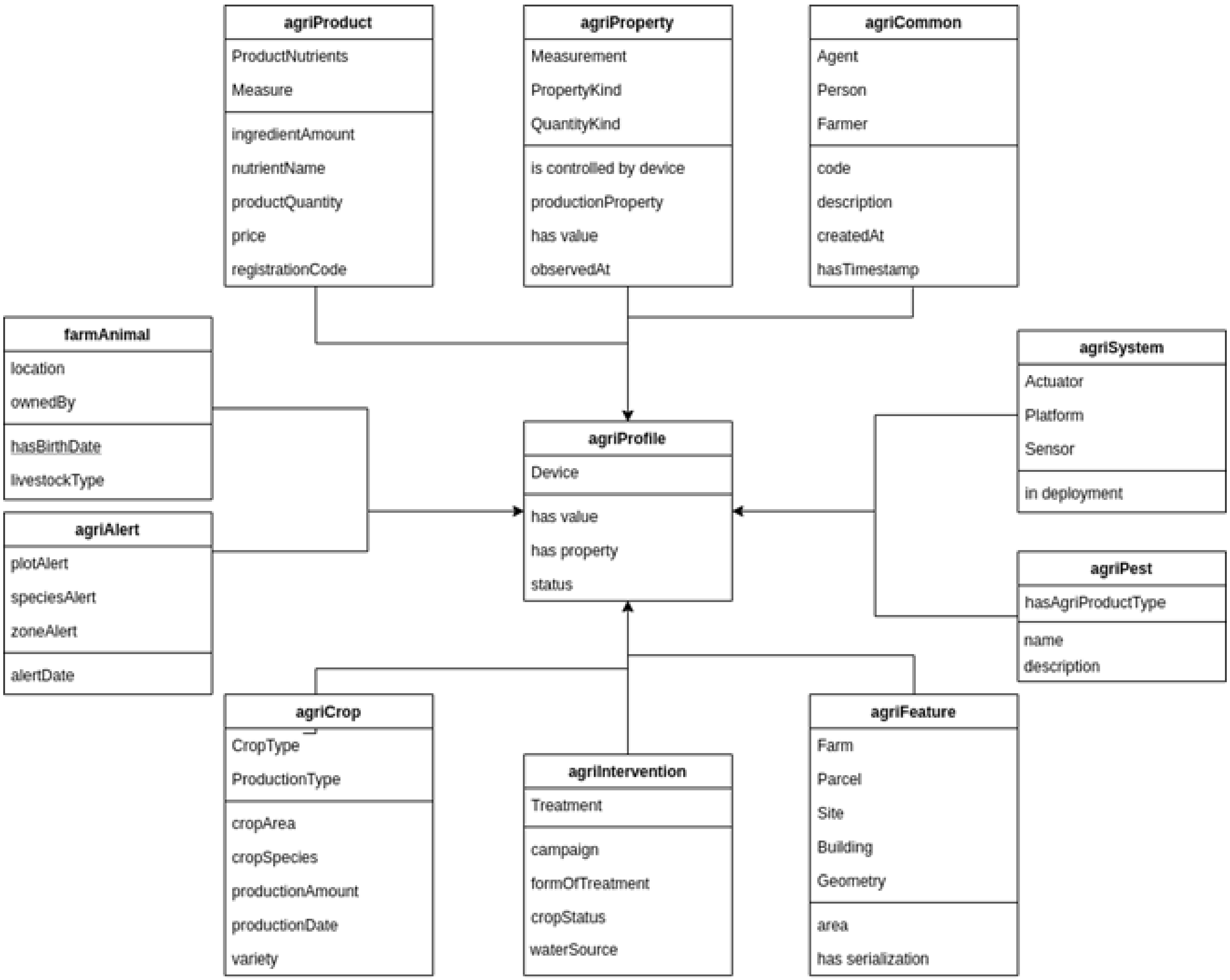

Figure 4. High-Level View of Domain-Specific Ontologies

8.3.1 Agriculture Commons Ontology

The initial release included a number of classes that were to be used by the other domain specific ontologies. After reviewing the whole Information Model, it was observed that such classes define more generic concepts and should be elevated to the cross-domain layer. These classes are *Agent, Person, Organization* and *Role*.

8.3.2 Agriculture Features Ontology

The first version encapsulated all the classes that define geometrical objects, such as lines, curves and polygons. The idea behind this was to model farms, plots and management zones. Though, these features may be used in more generic concepts, like a point can be used for coordinates independent of agricultural operations. So, in the second release of the AIM the following classes have been moved to the cross-domain layer: *Curve, Geometry, GeometryCollection, Line, LineString, LinearRing, MultiCurve, MultiLiveString, MultiPoint, MultiPolygon, MultiSurface, Point, Polygon, Polyhedral Surface, Surface, TIN* (Triangulated irregular network) and *Triangle*.

8.3.3 Agriculture Properties Ontology

This is another ontology that after the revision was found to contain terms that are more generic and subsequently appear in more models than first scheduled and expected. Such is the case of the classes *PropertyKind, QuantityKind* and the latter's subclasses: *Acceleration, Compressibility, Concentration, Density, Dimensionless, Distance, EnergyDensity, EnergyFlux, MassPerTimePerArea, Power, RadianceExposure, SpecificEntropy, StressOrPressure, SurfaceDensity, Temperature, ThermalConductivity, VelocityOrSpeed.* Same thing occurs with the basic class FeatureOfInterest, which is used in modelling numerous concepts that concern observations. Furthermore, several individuals have been also moved to the cross-domain layer: *AcousticsScheme, HeatScheme, MechanicsScheme, ElectricityAndMagnetismScheme, PhysicalChemistryAndMolecularPhysicsScheme, LightAndRelatedElectromagneticRadiationsScheme, SolidStatePhysicsScheme, SpaceAndTimeScheme, acceleration, carbonContent, compatibility, concentration, density, distance, energyFlowRate, fraction, length, massPerTimePerArea, power, pressure, property, radianceExposure, scalarProperty, soundEnergyDensity, soundIntensity, soundPower, soundPressure, surfaceDensity, temperature, vapourCompressibility, vapourPressure, vectorProperty, velocity.*

8.3.4 Agriculture System Ontology

So far, changes on domain specific ontologies have been related to moving generic terms that have been initially defined in an agri-specific ontology to the cross-domain layer. Same thing happens with this model. All the following object properties have been moved: deployed on platform, deployed system, generalQuantityKind, has deployment, has subsystem, host, in deployment, is hosted by, propertyType, skos:inScheme. Several classes have been also moved after the revision: *System, Deployment, Platform, Energy, Rotational Speed*. Lastly, all individuals have been moved to the cross-domain layer: *AcousticsScheme, MechanicsScheme, PeriodicAndRelatedPhenomenaScheme, SpaceAndTimeScheme, criticalBuildUpScheme, criticalTorsionalSpeed, criticalWhirlingSpeed, property, pseudovectorProperty, rotationalSpeed, synchronousPullOutTorque, synchronousSpeed* and *torque.*

8.4 Pilot-Specific ontologies

Apart of the domain-specific layer of AIM that covers the need for a more specific approach, it has been required multiple times to define more ontologies and modules which consist of terms and properties that are not defined in any well-known ontology and are still needed so that our pilots can integrate their systems to DEMETER and comply with AIM. Complementary to the domain-specific ontology that covers the needs identified by the requirement analysis we had performed, we should facilitate the integration and compliance of our pilots to AIM by all means. So, we started creating extensions of the domain-specific ontologies to provide tailor-made ontologies to some pilots that handle concepts that have never been defined before in any known ontology. The aforementioned extensions concern numerous operations that happen in the whole agricultural ecosystem from farm activities (e.g., livestock feeding) to tracking (e.g., transport conditions) and decision making (KPI indication). Soon, the number of extensions became increasingly bigger and the ontologies should import from higher layers than the domain-specific. The number and complexity of the required modules, as well as their implementation, led to the design decision of creating a new layer to represent these new ontologies that are quite independent in the way the terms are defined inside. The following subsection present completed ontologies of the pilot-specific layer.

8.4.1 Field Operations Ontology

This module covers the need of Pilots in Cluster 2, as well as Pilot 5.1 for an ontology that models a vehicle that operates on field, including both technical aspects, such as specifications or the trajectory of the vehicles, as well as more qualitative features such as the driver's performance. It imports from the Agriculture Commons ontology.

The classes defined in the ontology are *VehicleOperator, DriverBehaviour* and *VehicleTrajectory*. The model also re-uses *Vehicle* class from FIWARE.

The data properties match the object to a value, instead of other objects. Such defined properties in this ontology are *breaking, fuelConsumption, driverBehaviourValue, trajectoryDuration, trajectoryDistance, trajectoryAverageSpeed.* We also import *speed* property from FIWARE ontology.

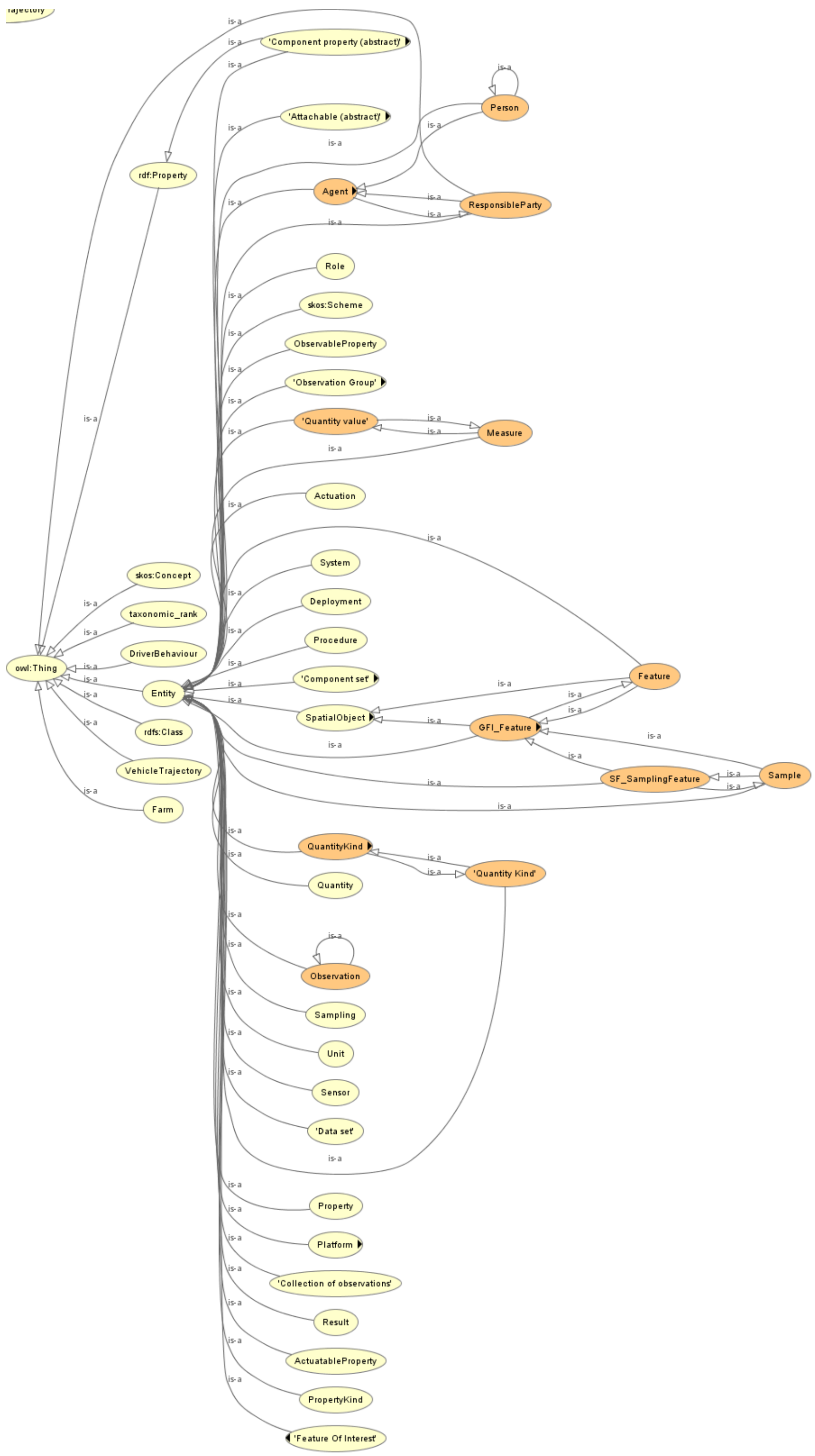

Figure 5. Field Operations Ontology Overview

8.4.2 Poultry Feeding Ontology

This ontology models the observation of the quality of poultry feeding at some point in time. It also includes information about the flock and silos used for storage. It imports and extends the Farm Animals ontology, tailored for pilots in cluster 4 and specifically in Pilots 4.4 and 5.4

The classes defined inside the module are *AnimalFeeding and Silos.* The data properties match the object to a value, instead of other objects. Such defined properties in this ontology are *silosFoodDensity, silosVolume, silosEmptyDistance, silosFoodType* which are related to storage information and *flockAverageAge* and *animalFeedingQuality* that describe the animals in observation*.*

This is a flat ontology, just extending the model without reusing terms from other well-known ontologies.

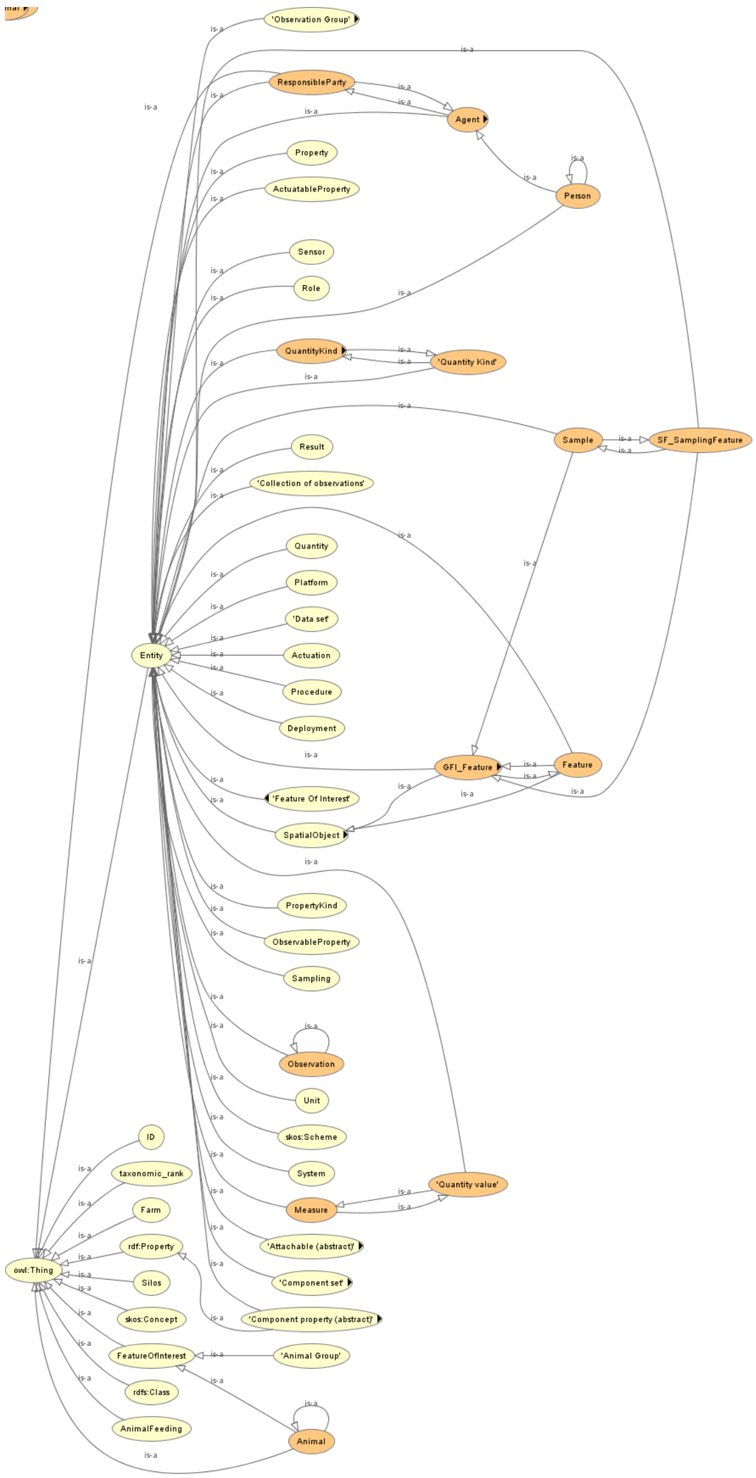

Figure 6. Poultry Feeding Ontology Overview

8.4.3 Stress Recognition Ontology

This ontology models the evaluation of the stress level of an animal group at some point in time and the instructions to be followed depending on the observation. The interested pilots are those of Cluster 4 and Pilots 5.2 and 5.4. It imports from a number of AIM domain specific ontologies: Agriculture System, Agriculture Product, Agriculture System and Poultry Feeding extension.

The module imports all required classes, so there are only data properties to be defined inside. These are *animalRawSound, extractedFeaturesFromSound, airHumidity, airflow, lightIntensity, airCO2, powerLose* and the output property: *stressLevel*

This module also uses no other known ontologies, though re-uses numerous AIM modules, as described above.

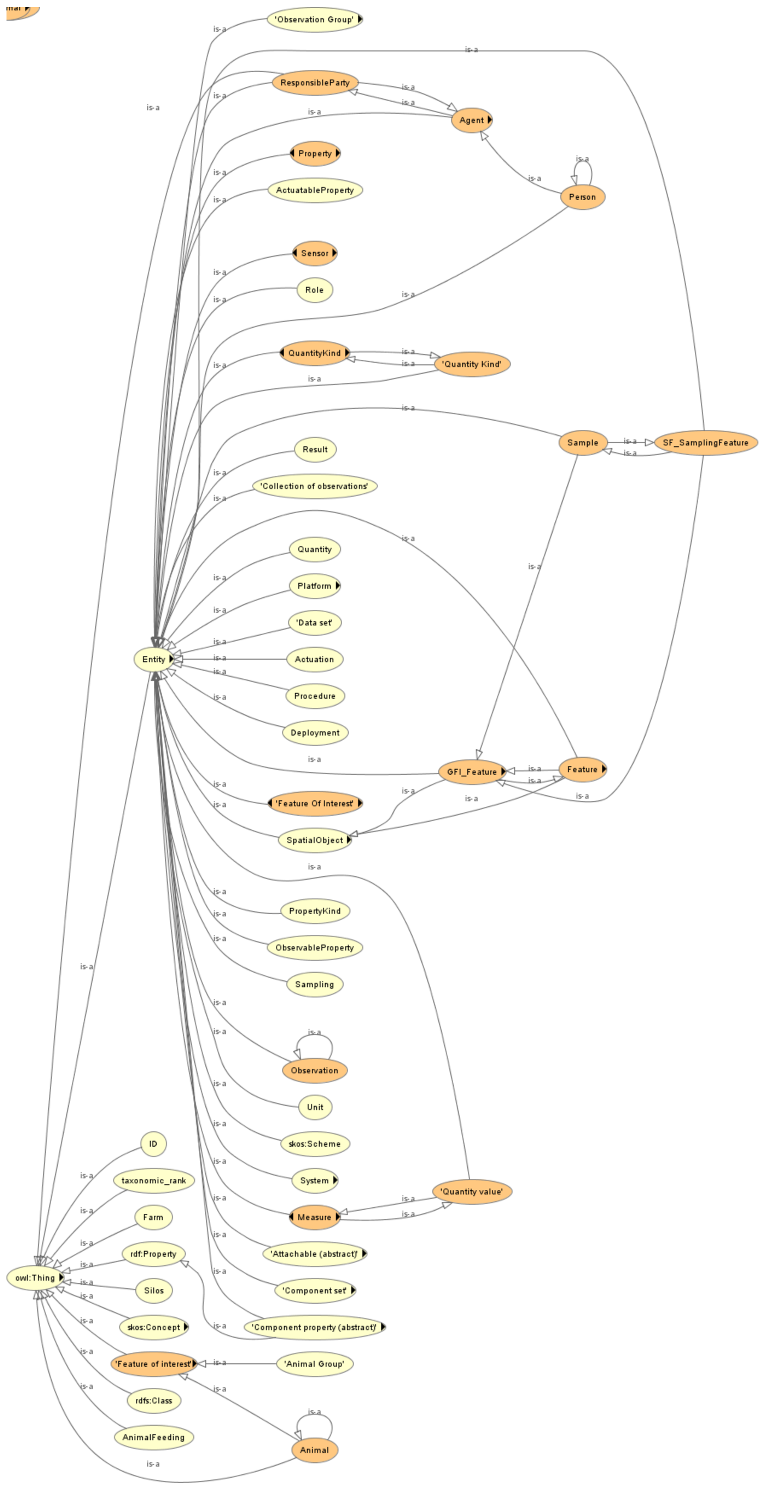

Figure 7. Stress Recognition Ontology Overview

8.4.4 Transport Condition Ontology

This module was tailored to Pilot 5.4, while being implemented, and models the evaluation of the conditions of transport of a poultry flock at some point in time based on their bio and health metrics. It imports terms from the Agriculture Product ontology.

The ontology defines the classes *Producer, PoultryProduct* and *Transport* and the object property *placeOfProduction*. There are also some data properties defined in this ontology. These are *certificates, mhr, poultryType, transportCondition* and *packageID*.

This is a flat ontology, just defining a whole new model without reusing terms from other well-known ontologies.

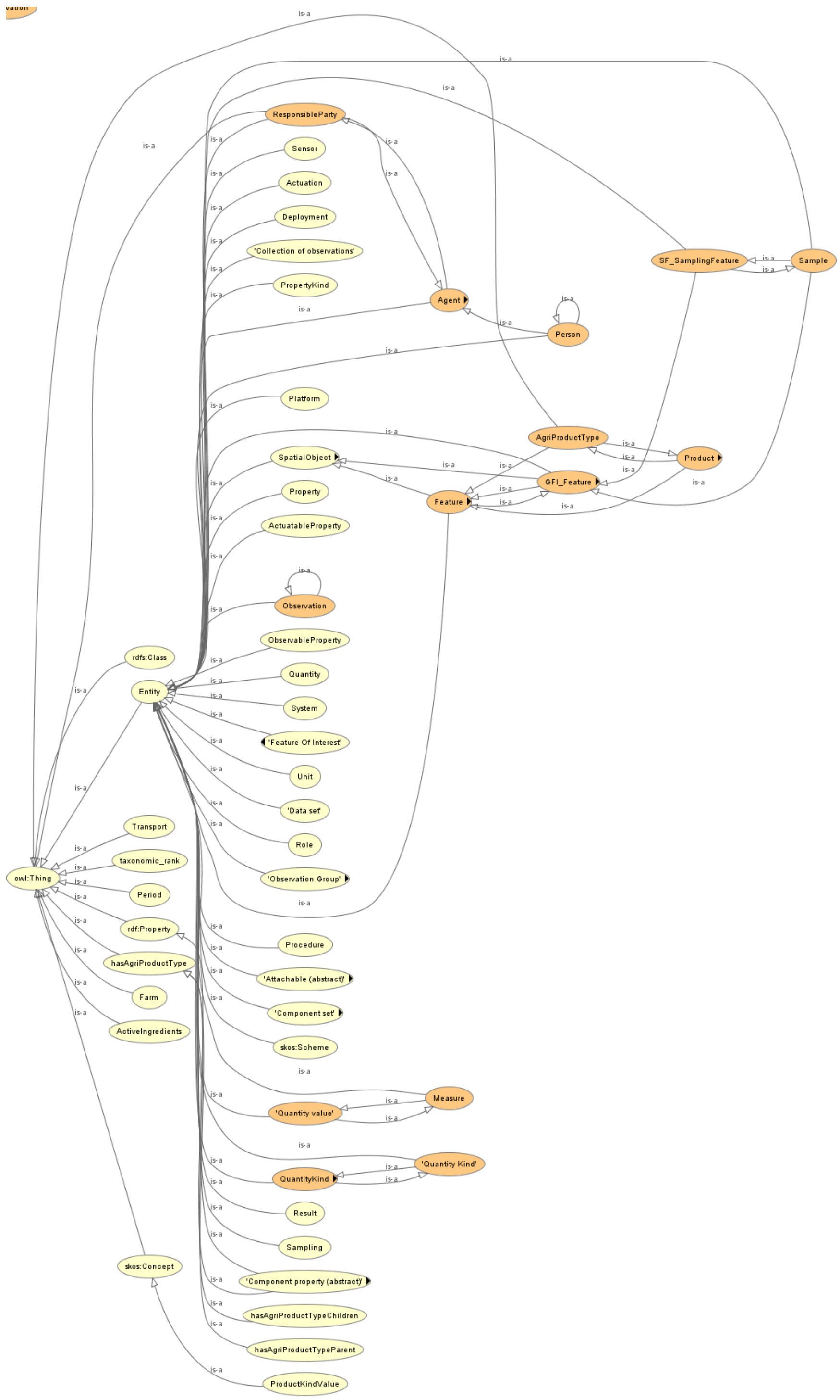

Figure 8. Transport Condition Ontology Overview

8.4.5 Nutrient Monitor Ontology

This module models information that concerns crop traits like morphological or physiological condition. It imports directly from the agriCommon ontology. The current version has emerged from two initially distinct ontologies, Nutrient Monitor and Vegetation Indices, used by Pilots 1.3, 1.4, 3.2 and 3.4.

The ontology consists of the basic class *CropTrait* that encapsulates the object properties that follow and the main classes *MorphologicalCondition* and *PhysiologicalCondition* that enclose the crop information. There are the respective object properties *hasMorphologicalCondition* and *hasPhysiologicalCondition* that relate the crop to the individuals like *leafNitrogen, leafLength, leafAnatomy* and *woodCarbon12*. There are also code lists that define types of leaf anatomy, specifically the *mesomorphic* and *xeromorphic* kinds.

Also, a number of data properties have been added, concerning more crop traits and vegetation indices: *ndvi*, *biomass*, *nitrogenContent, nitrogenConcentration*, *phosphorusUptake* and *potassiumUptake*.

8.4.6 KPI Indicators Ontology

This module is tailored to the needs of Pilot 2.4. The kpiIndicator.ttl is an ontology which is consisted of the following classes: *Entity, Farm, rdf:Property, rdfs:Class, ResponsibleParty, skos:Concept, skos:ConceptScheme, taxonomic_rank, ActuatableProperty, Actuation, Agent (Organization, Farm holding, Person, Farmer and Person), Attachable (abstract), Observation, Slice, Collection of observations, Component property (abstract), Attribute property, Coded property, Dimension property, Dimension property, Measure property, Component set, Component specification, Data structure definition, Slice key, Data set, Deployment, Feature, Feature Of Interest, GFI_Feature, GFI_DomainFeature, Feature Of Interest, Sample, SF_SamplingFeature, ObservableProperty, KpiIndicator, Observation, Observation, KpiIndicatorValue, Observation Group, Slice, Person, Farmer, Person, Platform, Procedure, Property, PropertyKind, Quantity, Quantity Kind, KpiIndicator, Quantity value, QuantityKind, Acceleration, Compressibility, Concentration, Density, Dimensionless, Distance, Energy, EnergyDensity, EnergyFlux, MassPerTimePerArea, Power, RadianceExposure, RotationalSpeed, SpecificEntropy, StressOrPressure, SurfaceDensity, Temperature, ThermalConductivity, VelocityOrSpeed, ResponsibleParty, Result, Role, Sample, Sampling, Sensor, SF_SamplingFeature, skos:Scheme, SpatialObject, Feature, Geometry, Geometry, Curve, Line String, Line, Linear Ring, Geometry Collection, Multi Curve, Multi Line String, Multi Point, Multi Surface, Multi Polygon, Point, Surface, Polygon, Triangle, Polyhedral Surface, Triangulated Irregular Network, Point, GFI_Feature, GFI_DomainFeature, Feature Of Interest, Sample, SF_SamplingFeature, System, Unit, Measure, Farm, rdf:Property, Component property (abstract), Attribute property, Coded property, Dimension property, Dimension property, Measure property, rdfs:Class, ResponsibleParty, skos:Concept, Sector, skos:ConceptScheme, taxonomic_rank.*

Object Properties are relationships defined between class objects. kpiIndicator.ttl is consisted of the following object properties: *agroVocConcept, eppoConcept, has result, isResultOf, manages farm, Property (contains, within), Relationship: component (dimension, measure), component attachment, component specification, concept, data set, deployed on platform, deployed system, has deployment, has feature of interest, has input, has output, has quantity kind, has subsystem, has ultimate feature of interest, hasGeometry, hasProperty, host, implemented by, implements, in deployment, is feature of interest of, is hosted by, isPropertyOf, location, made observation, madeBySensor, member, member observation, observation, observation group (slice), observedProperty, observes, phenomenon time, quantity value, slice key, slice structure, unit, used procedure, and sampledFeature, sector.*

Likewise, Data Properties match an object to a value and not another object. These are the *owl:topDataProperty*, which is consisted of the *A relation to express the name of an entity (e.g.,animal), altitude, begin of real-world phenomenon, category, code, dataProvider, description, description, end of real-world phenomenon, entityVersion, has description, has serialization(asWKT), has simple result, has timestamp (createdAt, generatedAtTime), invalidatedAtTime, latitude, longitude, modifiedAt, name (name, alternateName), notes, numeric value, order, Password, price, referenceValue, result time, source, type and User Name.*

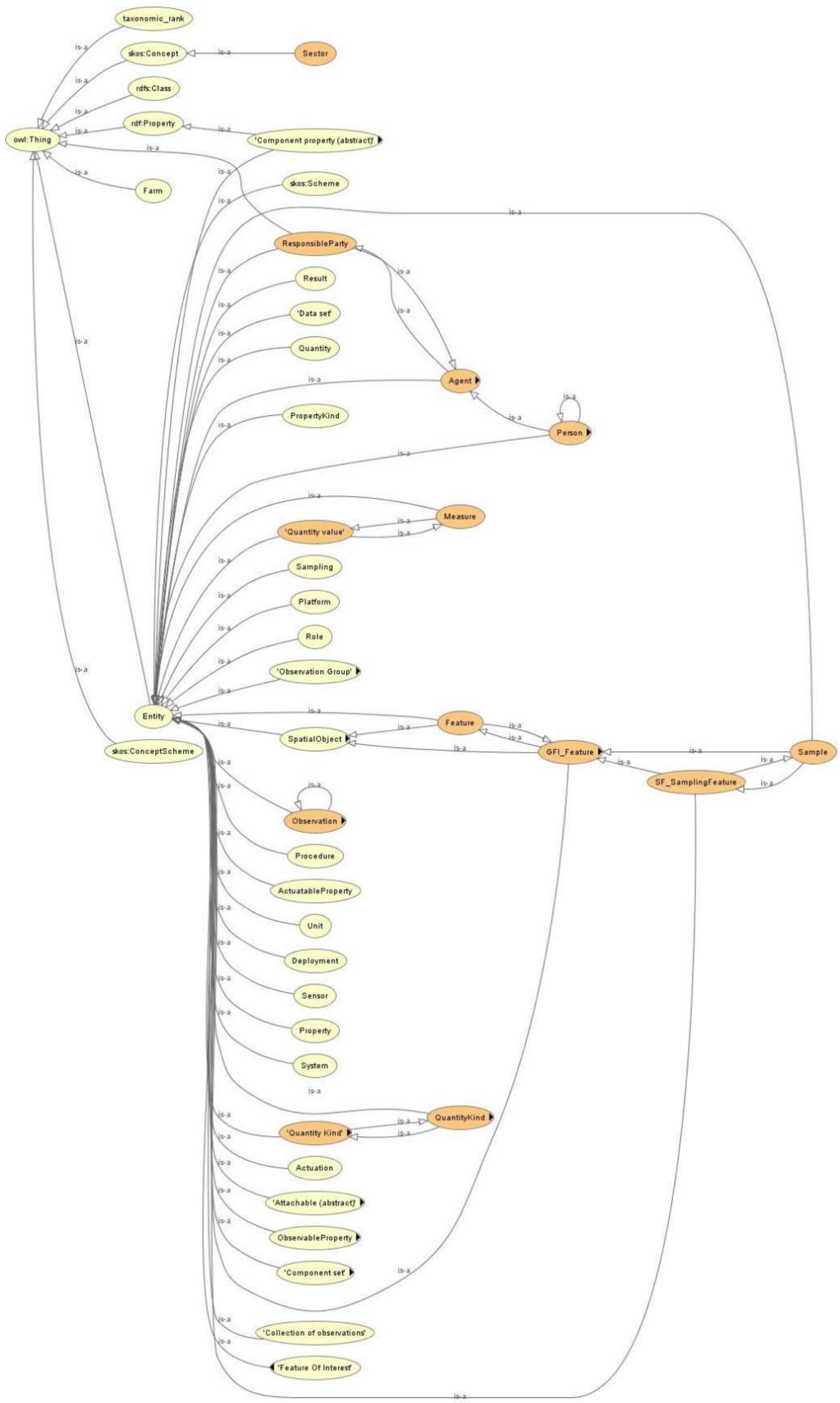

Figure 9. KPI Indicators Overview

8.4.7 Livestock Features Ontology

This module is created to be used across all pilots in Cluster 4 and 5. The livestockFeature.ttl is an ontology which consists of the following classes:*owl:Thing, Animal, Entity, ActuatableProperty, Actuation, Agent, Organization, Farm holding, Person, Farmer, Person, Attachable (abstract), Observation, Slice, Collection of observations, MilkQualityPrediction, Component property (abstract), Attribute property, Coded property, Dimension property, Dimension property, Measure property, Component set, Component specification, Data structure definition, Slice key, Data set, Deployment, Feature, Feature Of Interest, GFI_Feature, GFI_DomainFeature, Feature Of Interest, Sample, SF_SamplingFeature, ObservableProperty, Observation, Observation, HealthPrediction, Observation Group, Slice, Person, Farmer, Person, Platform, Procedure, Property, PropertyKind, Quantity, Quantity Kind, Quantity value, QuantityKind, Acceleration, Compressibility, Concentration, Density, Dimensionless, Distance, Energy, EnergyDensity, EnergyFlux, MassPerTimePerArea, Power, RadianceExposure, RotationalSpeed, SpecificEntropy, StressOrPressure, SurfaceDensity, Temperature, ThermalConductivity, VelocityOrSpeed, ResponsibleParty, Result, Role, Sample, Sampling, Sensor, SF_SamplingFeature, skos:Scheme, SpatialObject, Feature, Geometry, Geometry, Curve, Line String, Line, Linear Ring, Geometry Collection, Multi Curve, Multi Line String, Multi Point, Multi Surface, Multi Polygon, Point, Surface, Polygon, Triangle, Polyhedral Surface, Triangulated Irregular Network, Point, GFI_Feature, GFI_DomainFeature, Feature Of Interest, Sample, SF_SamplingFeature, System, Unit, Measure, Farm, FeatureOfInterest, Animal, Animal, FarmAnimalSpecies, Animal Group, ID, MilkProduct, PredictionMetric, rdf:Property, Component property (abstract), Attribute property, Coded property, Dimension property, Dimension property, Measure property, rdfs:Class, ResponsibleParty, skos:Concept, HealthStatus, QualityValue, skos:ConceptScheme, taxonomic_rank.*

The livestockFeature.ttl is consisted of the following object properties:*owl:topObjectProperty, agroVocConcept, calvedBy, eppoConcept, fedWith, has id, has member, has result, has_rank, includesAnimal, is located in, locatedAt, is location of, is member of, isResultOf, manages farm, ownedBy, predictionMetric, Property, contains, within, Relationship, component, dimension, measure, component attachment, component specification, concept, data set, deployed on platform, deployed system, has deployment, has feature of interest, has input, has output, has quantity kind, has subsystem, has ultimate feature of interest, hasGeometry, hasProperty, host, implemented by, implements, in deployment, is feature of interest of, is hosted by, isPropertyOf, location, made observation, madeBySensor, member, member observation, observation, observation group, slice, observedProperty, observes, phenomenon time, quantity value, slice key, slice structure, unit, used procedure, sampledFeature* and *siredBy*.

Likewise, Data Properties match an object to a value and not to another object. These properties are *owl:topDataProperty, A relation to express the name of an entity (e.g.,animal)., altitude, begin of real-world phenomenon, birthdate, breed, category, code, dataProvider, description, description, end of real-world phenomenon, entityVersion, has birth date, has death date, has description, has serialization, asWKT, has* *simple result, has timestamp, createdAt, generatedAtTime, healthCondition, invalidatedAtTime, invalidatedAtTime, healthCondition, invalidatedAtTime, latitude, legalID, livestockNumber, livestockType, longitude, modifiedAt, name, name, alternateName, notes, numeric value, order, Password, phenologicalCondition, predictionMetricProperty, ketosisAccuracy, ketosisFalsePositiveRate, ketosisPrecision, ketosisTruePositiveRate, lamenessAccuracy, lamenessFalsePositiveRate, lamenessPrecision, lamenessTruePositiveRate, mastitisAccuracy, mastitisFalsePositiveRate, mastitisPrecision, mastitisTruePositiveRate, processedAccuracy, processedFalsePositiveRate, processedPrecision, processedTruePositiveRate, rawAccuracy, rawFalsePositiveRate, rawPrecision, rawTruePositiveRate, price, relatedSource, reproductiveCondition, result time, sex, source, species, type, User Name, weight* and *welfareCondition.*

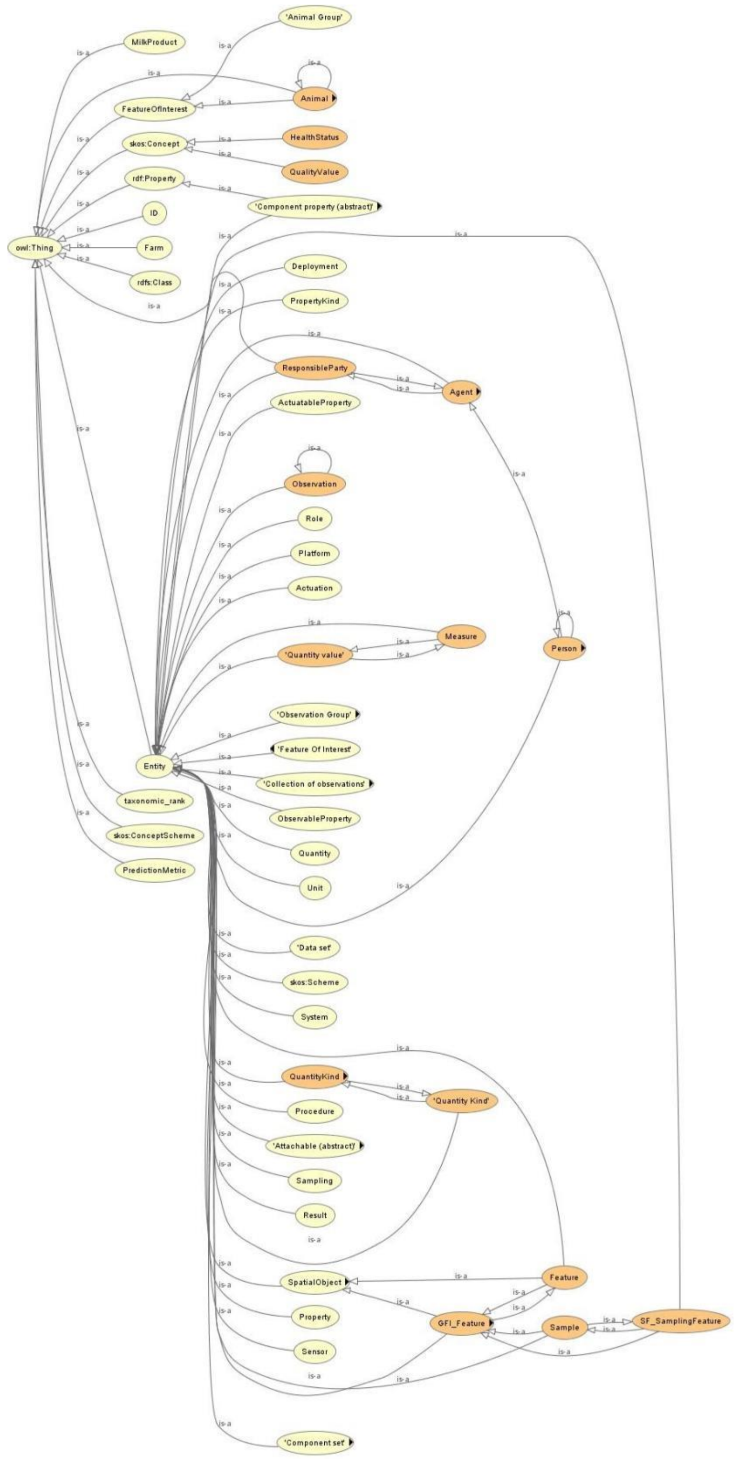

Figure 10. Livestock Features Ontology overview

8.5 Metadata Schema

The AIM Metadata Schema has undergone a major revision, in order to properly include Industrial Data Spaces (IDS) - as well as Data Quality Vocabulary (DQV) references. While maintaining the original URI (https://w3id.org/demeter/metadata), the previous Metadata Schema was discarded and replace with an entirely new version. The new Metadata Schema was created by reusing and aligning well-known ontologies and vocabularies. It builds up on DCAT 2.0 and uses a subset of the IDS Information Model to increase the expressivity of certain aspects of the AIM Metadata Schema. The usage of an IDS-compliant vocabulary also paves the way for potential future compatibility with the Industrial Data Space. Furthermore, the DEMETER Metadata Schema makes references to the W3C DQV to allow capturing data quality information determined by the WP2 Data Quality Components.

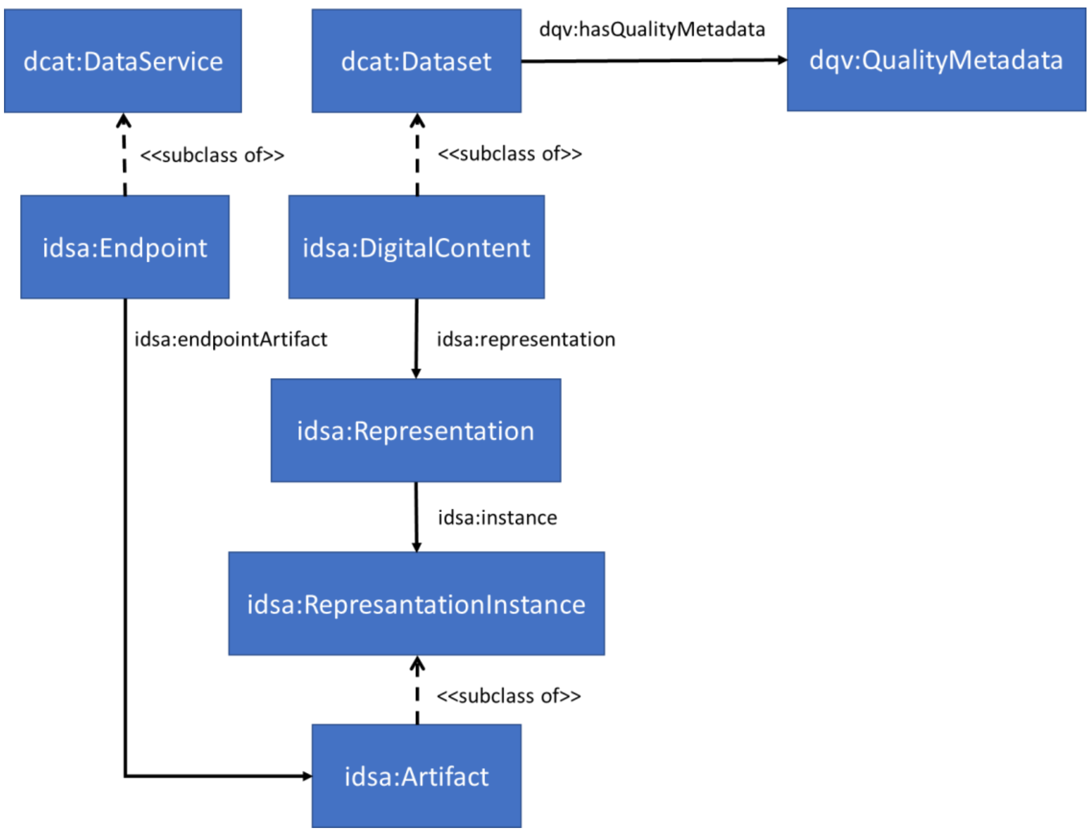

Figure 11. Core classes of the AIM Metadata Schema

The Metadata Schema is published as Turtle (TTL) file at:

https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/metadataschema

The Metadata Schema SHACL validation file is located at:

https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/metadataschema/-/tree/master/SHACL

Figure 11 shows the core classes of DCAT, IDS and DQV and how they were integrated within the AIM Metadata Schema. The idsa:Artifact represents a physical instance of a data set. It can represent, a physical file as well as the output of an Endpoint (e.g., a Webservice). Properties of idsa:Artifact refer to the physical nature of the data (timestamps, filesize, checksum) as shown in Figure 12. The idsa:Artifact is more specific than the dcat:Dataset or it's subclass idsa:DigitalContent. A dcat:Dataset (or it's sublclass idsa:DigitalConent) represent the properties of a data set which tend to be static across multiple batches from the same data source. These are properties such as the type of content (file format, schema), the frequency of updates, the temporal resolution and the spatial coverage. In general, the endpoint properties are shown in Figure 13.

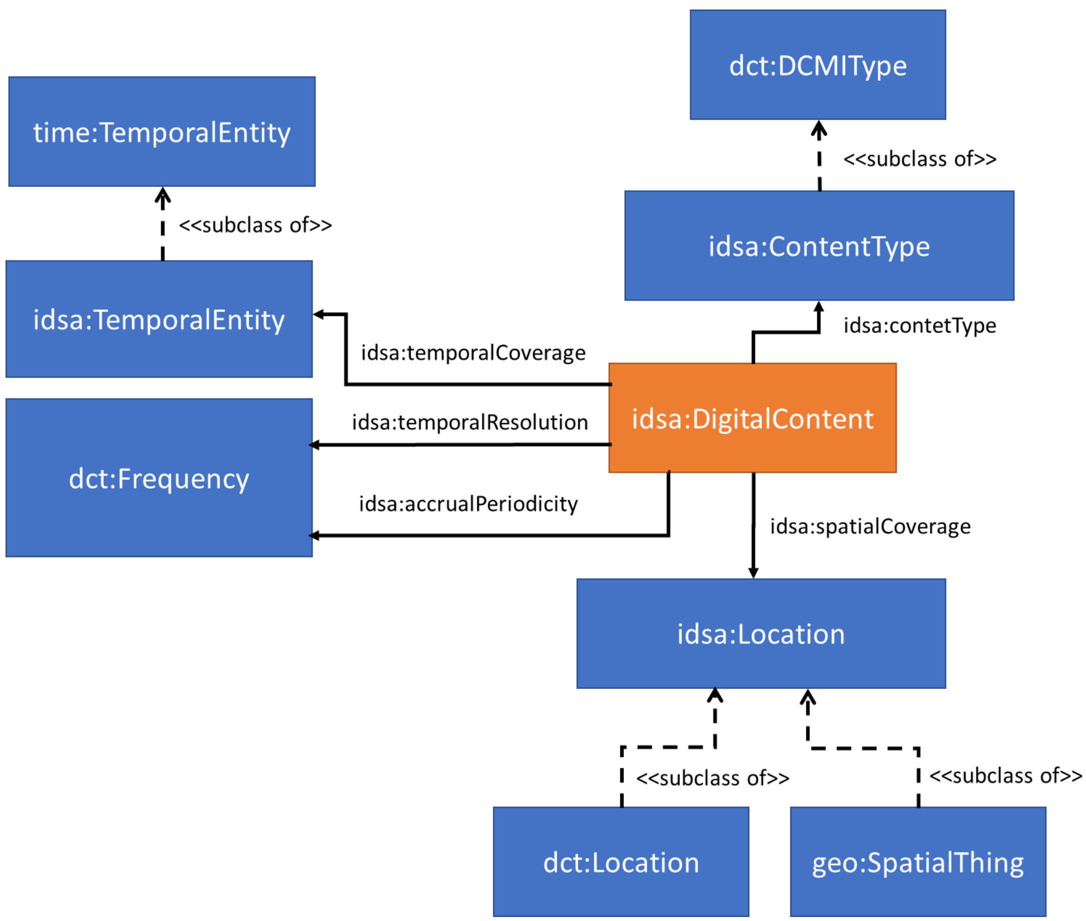

Figure 12. Digital Content properties in the AIM Metadata Schema

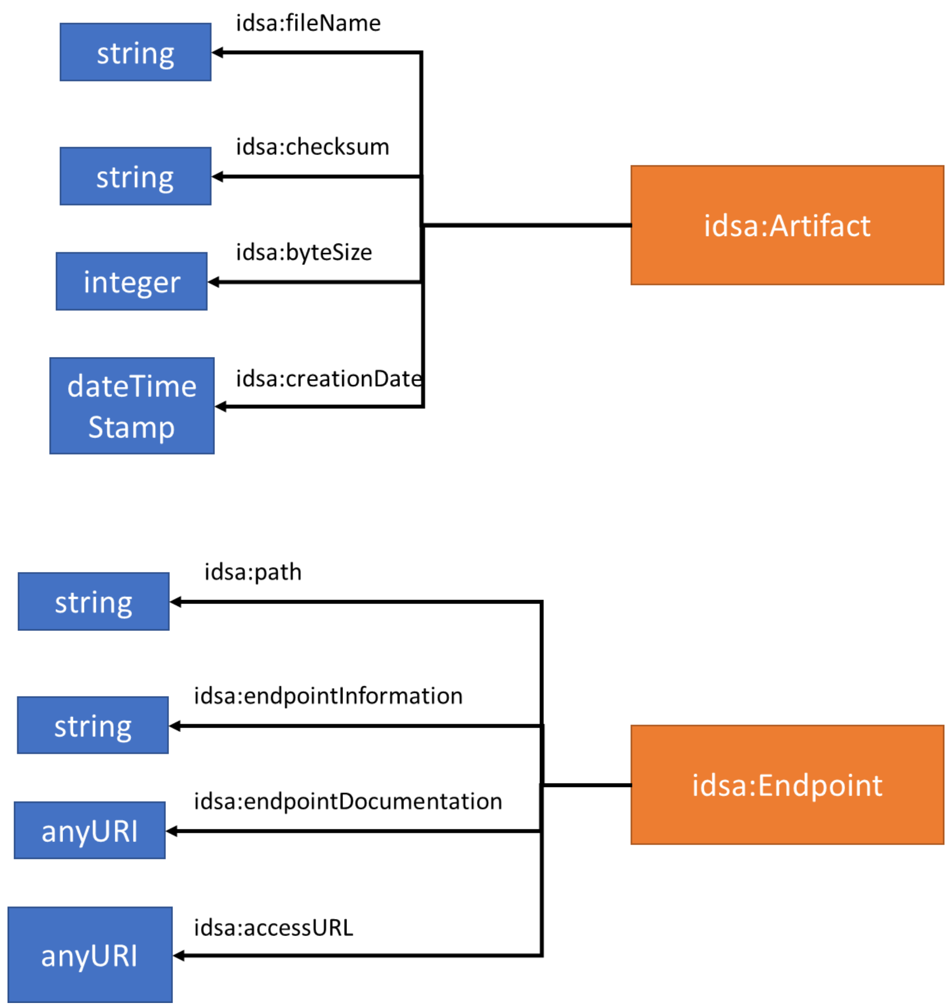

Figure 13. Artifact and Endpoint properties in the AIM Metadata Schema

IDS Information Model as Foundation

Via its comprehensive "Concern Hexagon" (please refer to Deliverable 2.1), the IDS Information Model provides a solid conceptual foundation for the AIM, which is also backed by a strong technical implementation that reuses all the other metadata standards identified as relevant for DEMETER in Deliverable 2.1 (DQV, PROV-O, etc.). For its latest version 4.0, it has been upgraded to DCAT 2. Plus, most recently, it has been following the profiling principle, which has been identified as crucial for the extensibility of the AIM above. To further make sure that the DEMETER AIM satisfies the requirements of DCAT-AP, the DEMETER AIM is formally defined as a DCAT profile using the Profiles vocabulary, and its relation to other existing DCAT profiles is being clarified. The resulting DCAT profile thus defines the general AIM metamodel elements to be referenced by any DEMETER generated datasets.

In particular, the DCAT-AP-DEMETER profile, which is provided by the DEMETER-profiled IDS Information Model, uses a general DCAT-DQV profile, and inherits the constraints required to use DQV with DCAT. Note DCAT-DQV itself inherits a DCAT-Datacube profile, a DCAT-PROV-O profile and a DCAT-DUV profile. While DCAT-PROV-O and DCAT-DUV are part of the meta-layer, DCAT-Datacube resides in the cross-domain layer. Hence there should be metadata profiles linked to cross-domain models to be able to provide a metadata schema optimised for each key sub-type of data. The profiles have been created by using stubs in DEMETER namespaces for each model and linking those stubs as profiles without extra constraints to the official standards.

By the same mechanism as explained so far, any further AIM data model module will have a declared DCAT profile derived detailing use of the specific AIM data elements, and any inherited conformance benefits derived from the alignment of that module with standard models. Such profiles will inherit conformance claims from the cross-domain ontologies they use. For example, many AIM modules will implement a profile of sosa:Observation – so it will be possible to control metadata requirements for Observations from a single point, but also to discover all datasets that contain data using the sosa:Observation model, without having to download the data and interrogate its data type hierarchies:

- DCAT-AP-DEMETER includes IDS constraints - and all DEMETER dataset metadata conform to IDS, or
- DCAT-AP-IDS is defined, and DCAT-AP-DEMETER-IDS is a subprofile to describe those datasets that conform to IDS model.

DEMETER will profile IDS where applicable and future IDS improvements, aligned with specific DEMETER needs, will facilitate the integration process.

Hence, similar to the approach taken in several recent EU initiatives such as the CYBELE project[4], DEMETER AIM metamodel will include the relevant properties to describe datasets and other digital content relevant to the agri-food sector with focus on quality aspects, and according to IDS constraints.

[4] https://www.cybele-project.eu/

European Union
European Regional
Development Fund

9 Revised Semantic Interoperability Support

The stated goal of the DEMETER AIM is to provide semantic interoperability with other existing systems and ontologies. The DEMETER AIM has been implemented by reusing terms from the most relevant ontologies and data models in the different areas relevant to support the final DEMETER applications, and using as a base the NGSI-LD meta-model and approach. DEMETER AIM defines a set of modules facilitating the scalability and maintainability of the model. Additionally, it includes alignments between key elements of these models in order to support the integration of existing datasets (which are based on them). Mostly (but not just) in the domain specific layer, the DEMETER AIM defines over a dozen modules reusing terms from SAREF4Agri, FIWARE Agrifood data models, FOODIE and other well-known ontologies. These allow AIM to be semantically interoperable with all these models and ontologies.

In this section, we provide updated tables regarding the semantic mappings of AIM to several of these well-known ontologies and vocabularies. Now, these for the most part are the same ontologies for which semantic mappings were presented in D2.1. However, here, the tables that were presented in D2.1 have been updated, in view of the changes to AIM that are described in the previous section and the updated AIM mappings to these ontologies are presented. In each subsection we give just the table with the semantic mappings and would refer the reader to the equivalent section in D2.1 for more details regarding the ontologies and system being mapped to.

We also include towards the end of this section, AIM mappings and interoperability with the EPPO model, which is new mapping that was not present in D2.1 and therefore that part is a full description of the interoperability, not just a table with the mapping.

9.1 Semantic Mapping to FIWARE

The following table maps the FIWARE AgriFood terms to the DEMETER AIM. The concepts mapped include classes, object and data properties. For each mapping the FIWARE and AIM terms are given, together with the type of entity that they map, the type of mapping being made and, finally, the AIM module in which the mapping is defined. We want to point out that wherever a term is being reused in AIM, then that term is also the AIM mapping itself and we denote this by "n/a" in the table.

Table 3. FIWARE AgriFood term mappings to DEMETER AIM

FIWARE term	Type	AIM mapping	mapping_type	AIM module
fiware:AgriCrop	class	saref4agri:Crop	equivalentClass	agriCrop
fiware:AgriFarm	class	saref4agri:Farm	equivalentClass	agriFeature
fiware:AgriParcel	class	saref4agri:Parcel	equivalentClass	agriFeature
fiware:AgriParcelOperation	class	foodie:Treatment	equivalentClass	agriIntervention

fiware:AgriProductType	class	foodie:Product	equivalentClass	agriProduct
fiware:Alert	class	foodie:Alert	equivalentClass	agriAlert
fiware:Animal	class	saref4agri:Animal	equivalentClass	farmAnimal
fiware:AgriGreenhouse	class	fiware:AgriParcel	subClassOf	agriFeature
fiware:WeatherForecast	class	saref:FeatureOfInterest	subClassOf	agriProperty
fiware:WeatherObserved	class	saref:FeatureOfInterest	subClassOf	agriProperty
fiware:hasAgriProductTypeChildren	class	fiware:hasAgriProductType	subClassOf	agriProduct
fiware:hasAgriProductTypeParent	class	fiware:hasAgriProductType	subClassOf	agriProduct
fiware:AgriParcelRecord	class	n/a	reused	agriProperty
fiware:AgriPest	class	n/a	reused	agriPest
fiware:hasAgriProductType	class	n/a	reused	agriProduct
fiware:hasAgriCrop	object_property	foodie:crop	equivalentProperty	agriFeature
fiware:hasOperator	object_property	foodie:operator	equivalentProperty	agriIntervention
fiware:operationHasAgriParcel	object_property	foodie:interventionPlot	equivalentProperty	agriIntervention
fiware:refDevice	object_property	saref:measurementMadeBy	equivalentProperty	agriProperty
fiware:hasAgriParcel	object_property	saref4agri:contains	subPropertyOf	agriFeature
fiware:hasAgriParcelChildren	object_property	saref4agri:contains	subPropertyOf	agriFeature
fiware:hasAgriParcelParent	object_property	saref4agri:isContainedIn	subPropertyOf	agriFeature
fiware:landLocation	object_property	fiware:location	subPropertyOf	agriFeature
fiware:locatedAt	object_property	saref4agri:isLocatedIn	subPropertyOf	farmAnimal
fiware:location	object_property	geosparql:hasGeometry	subPropertyOf	agriFeature
fiware:agroVocConcept	object_property	n/a	reused	agriCommon
fiware:calvedBy	object_property	n/a	reused	farmAnimal
fiware:cropHasAgriSoil	object_property	n/a	reused	agriCrop
fiware:fedWith	object_property	n/a	reused	farmAnimal
fiware:hasAgriFertiliser	object_property	n/a	reused	agriCrop
fiware:hasAgriPest	object_property	n/a	reused	agriCrop
fiware:hasAgriProductType	object_property	n/a	reused	agriProduct
fiware:hasAgriProductTypeChildren	object_property	n/a	reused	agriProduct

fiware:hasAgriProductTypeParent	object_property	n/a	reused	agriProduct
fiware:hasAgriSoil	object_property	n/a	reused	agriFeature
fiware:hasDevice	object_property	n/a	reused	agriProperty
fiware:ownedBy	object_property	n/a	reused	farmAnimal
fiware:recordHasAgriParcel	object_property	n/a	reused	agriProperty
fiware:refPointOfInterest	object_property	n/a	reused	agriProperty
fiware:siredBy	object_property	n/a	reused	farmAnimal
fiware:birthdate	data_property	saref4agri:hasBirthDate	equivalentProperty	farmAnimal
fiware:category	data_property	foodie:type	equivalentProperty	agriCommon
fiware:description	data_property	saref:hasDescription	equivalentProperty	agriCommon
fiware:description	data_property	foodie:description	equivalentProperty	agriCommon
fiware:legalID	data_property	foodie:livestockNumber	equivalentProperty	farmAnimal
fiware:name	data_property	saref4agri:hasName	equivalentProperty	agriCommon
fiware:species	data_property	foodie:livestockType	equivalentProperty	farmAnimal
fiware:status	data_property	foodie:status	equivalentProperty	agriIntervention
fiware:airTemperature	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:alternateName	data_property	fiware:name	subPropertyOf	agriCommon
fiware:atmosphericPressure	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:createdAt	data_property	saref:hasTimestamp	subPropertyOf	agriCommon
fiware:dateObserved	data_property	saref:hasTimestamp	subPropertyOf	agriProperty
fiware:dewPoint	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:endedAt	data_property	inspire-base:validTo	subPropertyOf	agriIntervention
fiware:illuminance	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:name	data_property	schema:name	subPropertyOf	agriCommon
fiware:observedAt	data_property	saref:hasTimestamp	subPropertyOf	agriProperty
fiware:operationType	data_property	foodie:type	subPropertyOf	agriIntervention
fiware:precipitation	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:pressureTendency	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:relativeHumidity	data_property	saref:hasValue	subPropertyOf	agriProperty

fiware:result	data_property	foodie:notes	subPropertyOf	agriIntervention
fiware:snowHeight	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:soilMoistureEc	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:soilMoistureVwc	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:soilTemperature	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:solarRadiation	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:startedAt	data_property	inspire-base:validFrom	subPropertyOf	agriIntervention
fiware:streamGauge	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:temperature	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:validFrom	data_property	inspire-base:validFrom	subPropertyOf	agriAlert
fiware:validTo	data_property	inspire-base:validTo	subPropertyOf	agriAlert
fiware:visibility	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:windDirection	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:windSpeed	data_property	saref:hasValue	subPropertyOf	agriProperty
fiware:address	data_property	n/a	reused	agriAlert
fiware:alertSource	data_property	n/a	reused	agriAlert
fiware:area	data_property	n/a	reused	agriFeature
fiware:breed	data_property	n/a	reused	farmAnimal
fiware:cropStatus	data_property	n/a	reused	agriFeature
fiware:data	data_property	n/a	reused	agriAlert
fiware:dataProvider	data_property	n/a	reused	agriCommon
fiware:dateIssued	data_property	n/a	reused	agriAlert
fiware:entityVersion	data_property	n/a	reused	agriCommon
fiware:harvestingInterval	data_property	n/a	reused	agriCrop
fiware:healthCondition	data_property	n/a	reused	farmAnimal
fiware:lastPlantedAt	data_property	n/a	reused	agriFeature
fiware:modifiedAt	data_property	n/a	reused	agriCommon
fiware:phenologicalCondition	data_property	n/a	reused	farmAnimal
fiware:plannedEndAt	data_property	n/a	reused	agriIntervention

fiware:plannedStartAt	data_property	n/a	reused	agriIntervention
fiware:plantingFrom	data_property	n/a	reused	agriCrop
fiware:quantity	data_property	n/a	reused	agriIntervention
fiware:relatedSource	data_property	n/a	reused	farmAnimal
fiware:reportedAt	data_property	n/a	reused	agriIntervention
fiware:reproductiveCondition	data_property	n/a	reused	farmAnimal
fiware:root	data_property	n/a	reused	agriProduct
fiware:severity	data_property	n/a	reused	agriAlert
fiware:sex	data_property	n/a	reused	farmAnimal
fiware:source	data_property	n/a	reused	agriCommon
fiware:subCategory	data_property	n/a	reused	agriAlert
fiware:waterSource	data_property	n/a	reused	agriIntervention
fiware:wateringFrequency	data_property	n/a	reused	agriCrop
fiware:weatherType	data_property	n/a	reused	agriProperty
fiware:weight	data_property	n/a	reused	farmAnimal
fiware:welfareCondition	data_property	n/a	reused	farmAnimal

9.2 Semantic Mapping to Saref4Agri

The following table maps the Saref4Agri terms to the DEMETER AIM. The concepts mapped include classes, object and data properties. For each mapping the Saref4Agri and AIM terms are given, together with the type of entity that they map, the type of mapping being made and, finally, the AIM module in which the mapping is defined. We want to point out that wherever a term is being reused in AIM, then that term is also the AIM mapping itself and we denote this by "n/a" in the table.

Table 4. Mapping of Saref4Agri terms to AIM

Saref4Agri/Saref term	**type**	**AIM mapping**	**mapping_type**	**AIM module**
saref4agri:Animal	class	inspire-af:FarmAnimalSpecies	equivalentClass	farmAnimal
saref4agri:Animal	class	fiware:Animal	equivalentClass	farmAnimal
saref4agri:Crop	class	fiware:AgriCrop	equivalentClass	agriCrop

saref4agri:Crop	class	foodie:CropSpecies	equivalentClass	agriCrop
saref4agri:Farm	class	inspire-af:Holding	equivalentClass	agriFeature
saref4agri:Farm	class	fiware:AgriFarm	equivalentClass	agriFeature
saref4agri:Parcel	class	foodie:Plot	equivalentClass	agriFeature
saref4agri:Parcel	class	fiware:AgriParcel	equivalentClass	agriFeature
saref:FeatureOfInterest	class	sosa:FeatureOfInterest	equivalentClass	agriProperty
saref:Measurement	class	sosa:Observation	equivalentClass	agriProperty
saref:Property	class	foodie:PropertyTypeValue	equivalentClass	agriProperty
saref:Property	class	qu:QuantityKind	equivalentClass	agriProperty
saref:Property	class	ssn:Property	equivalentClass	agriProperty
saref:Sensor	class	sosa:Sensor	equivalentClass	agriSystem
saref:Temperature	class	qu-dim:Temperature	equivalentClass	agriProperty
saref:UnitOfMeasure	class	qudt:Unit	equivalentClass	agriProperty
saref4agri:Animal	class	saref:FeatureOfInterest	subClassOf	farmAnimal
saref4agri:AnimalGroup	class	saref:FeatureOfInterest	subClassOf	farmAnimal
saref4agri:Building	class	geosparql:Feature	subClassOf	agriFeature
saref4agri:BuildingSpace	class	geosparql:Feature	subClassOf	agriFeature
saref4agri:Crop	class	saref:FeatureOfInterest	subClassOf	agriCrop
saref4agri:EatingActivitySensor	class	saref:Sensor	subClassOf	agriSystem
saref4agri:Farm	class	geosparql:Feature	subClassOf	agriFeature
saref4agri:FarmHolding	class	schema:Organization	subClassOf	agriCommon
saref4agri:Farmer	class	foaf:Person	subClassOf	agriCommon
saref4agri:MilkingSensor	class	saref:Sensor	subClassOf	agriSystem
saref4agri:MovementActivitySensor	class	saref:Sensor	subClassOf	agriSystem
saref4agri:Parcel	class	geosparql:Feature	subClassOf	agriFeature
saref4agri:Pluviometer	class	saref:Sensor	subClassOf	agriSystem
saref4agri:Soil	class	saref:FeatureOfInterest	subClassOf	agriProperty
saref4agri:SoilTensiometer	class	saref:Sensor	subClassOf	agriSystem
saref4agri:Thermometer	class	saref:Sensor	subClassOf	agriSystem

saref4agri:WateringGun	class	saref:Actuator	subClassOf	agriSystem
saref4agri:WateringSystem	class	ssn:System	subClassOf	agriSystem
saref4agri:WateringValve	class	saref:Actuator	subClassOf	agriSystem
saref4agri:WeatherStation	class	ssn:System	subClassOf	agriSystem
saref4agri:WeatherStation	class	saref:Sensor	subClassOf	agriSystem
saref4agri:WeightSensor	class	saref:Sensor	subClassOf	agriSystem
saref:Actuator	class	saref:Device	subClassOf	agriSystem
saref:Device	class	ssn:System	subClassOf	agriSystem
saref:Humidity	class	saref:Property	subClassOf	agriProperty
saref:Sensor	class	saref:Device	subClassOf	agriSystem
saref:Temperature	class	saref:Property	subClassOf	agriProperty
saref4agri:ID	class	n/a	reused	farmAnimal
saref:hasFeatureOfInterest	object_property	sosa:hasFeatureOfInterest	equivalentProperty	agriProperty
saref:hasProperty	object_property	ssn:hasProperty	equivalentProperty	agriProperty
saref:isFeatureOfInterestOf	object_property	sosa:isFeatureOfInterestOf	equivalentProperty	agriProperty
saref:isMeasuredIn	object_property	qudt:unit	equivalentProperty	agriProperty
saref:isPropertyOf	object_property	ssn:isPropertyOf	equivalentProperty	agriProperty
saref:makesMeasurement	object_property	sosa:madeObservation	equivalentProperty	agriProperty
saref:measurementMadeBy	object_property	sosa:madeBySensor	equivalentProperty	agriProperty
saref:measurementMadeBy	object_property	fiware:refDevice	equivalentProperty	agriProperty
saref:relatesToProperty	object_property	foodie:propertyType	equivalentProperty	agriProperty
saref:relatesToProperty	object_property	sosa:observedProperty	equivalentProperty	agriProperty
saref4agri:contains	object_property	geosparql:sfContains	subPropertyOf	agriFeature
saref4agri:generates	object_property	saref:hasProperty	subPropertyOf	agriCrop
saref4agri:isContainedIn	object_property	geosparql:sfWithin	subPropertyOf	agriFeature
saref4agri:receives	object_property	saref:hasProperty	subPropertyOf	agriCrop
saref4agri:hasID	object_property	n/a	reused	farmAnimal
saref4agri:hasMember	object_property	n/a	reused	farmAnimal
saref4agri:isLocatedIn	object_property	n/a	reused	farmAnimal

saref4agri:isLocationOf	object_property	n/a	reused	farmAnimal
saref4agri:isMemberOf	object_property	n/a	reused	farmAnimal
saref4agri:managesFarm	object_property	n/a	reused	agriCommon
saref:controlsProperty	object_property	n/a	reused	agriProperty
saref:isControlledByDevice	object_property	n/a	reused	agriProperty
saref:isMeasuredByDevice	object_property	n/a	reused	agriProperty
saref:measuresProperty	object_property	n/a	reused	agriProperty
saref:relatesToMeasurement	object_property	n/a	reused	agriProperty
saref4agri:hasBirthDate	data_property	fiware:birthdate	equivalentProperty	farmAnimal
saref4agri:hasHarvestDate	data_property	foodie:productionDate	equivalentProperty	agriCrop
saref4agri:hasName	data_property	fiware:name	equivalentProperty	agriCommon
saref:hasDescription	data_property	foodie:description	equivalentProperty	agriCommon
saref:hasDescription	data_property	fiware:description	equivalentProperty	agriCommon
saref:hasTimestamp	data_property	sosa:resultTime	equivalentProperty	agriCommon
saref:hasValue	data_property	sosa:hasSimpleResult	equivalentProperty	agriProperty
saref4agri:hasHarvestDate	data_property	owl:topDataProperty	subPropertyOf	agriCrop
saref4agri:hasPlantDate	data_property	owl:topDataProperty	subPropertyOf	agriCrop
saref4agri:hasDeathDate	data_property	n/a	reused	farmAnimal

9.3 Semantic Mapping to ADAPT

The following table maps the Saref4Agri terms to the DEMETER AIM. The concepts mapped include classes, object and data properties. For each mapping the Saref4Agri and AIM terms are given, together with the type of entity that they map, the type of mapping being made and, finally, the ADAPT module in which the ADAPT concept is defined. All of these mappings are implemented in the aim-adapt.ttl alignment file.

Table 5. Mapping of ADAPT terms to AIM

ADAPT term	Type	AIM mapping	mapping_type	ADAPT module
adapt:Farm	class	saref4agri:Farm	equivalentClass	Grower
adapt:Field	class	saref4agri:Parcel	equivalentClass	Grower
adapt:Grower	class	saref4agri:Farmer	subClassOf	Grower

adapt:CropZone	class	foodie:CropSpecies	equivalentClass	Grower
adapt:Location	class	geo:Feature	subClassOf	Grower
adapt:Facility	class	geo:Feature	subClassOf	Grower
adapt:FieldBoundary	class	geo:Feature	subClassOf	FieldBoundary
adapt:Product	class	foodie:Product	owl:equivalentClass	aim-adapt
adapt:CropProtectionProduct	class	foodie:Product	subClassOf	Product
adapt:CropVarietyProduct	class	foodie:Product	subClassOf	Product
adapt:CropNutritionProduct	class	foodie:Product	subClassOf	Product
adapt:FertilizerProduct	class	foodie:Product	subClassOf	Product
adapt:IngredientUse	class	foodie:ActiveIngredients	owl:equivalentClass	Product
adapt:UnitOfMeasure	class	qudt:Unit	owl:equivalentClass	UOM
adapt:UnitOfMeasureDimensionEnum	class	qudt:QuantityKind	owl:equivalentClass	UOM
adapt:Obs	class	sosa:Observation	rdfs:subClassOf	Observations
adapt:ObsCollection	class	sosa:ObservationCollection	rdfs:subClassOf	Observations
adapt:OMCode	class	sosa:ObservableProperty	rdfs:subClassOf	Observations
adapt:Place	class	sosa:FeatureOfInterest	rdfs:subClassOf	Observations
adapt:Shape	class	geo:Geometry	owl:equivalentClass	Shape
adapt:Id	data_property	foodie:code	owl:equivalentProperty	Grower
adapt:Description	data_property	foodie:description	owl:equivalentProperty	Grower
adapt:Position	object_property	geo:hasGeometry	rdfs:subPropertyOf	Grower
adapt:Area	data_property	fiware:area	owl:equivalentProperty	Grower
adapt:BoundingRegion	object_property	geo:hasGeometry	rdfs:subPropertyOf	Grower
adapt:Notes	data_property	foodie:notes	owl:equivalentProperty	Grower
adapt:SpatialExtent	object_property	geo:hasGeometry	owl:equivalentProperty	Observations

9.4 Semantic Mapping to INSPIRE and FOODIE

The following table maps the INSPIRE terms and the FOODIE terms to the DEMETER AIM. The concepts mapped include classes, object and data properties. For each mapping the Saref4Agri and AIM terms are given, together with the type of entity that they map, the type of mapping being made and, finally, the AIM module in which the mapping is defined. We want to point out that wherever a term is being reused in AIM, then that term is also the AIM mapping itself and we denote this by "n/a" in the table.

Table 6. Mapping of INSPIRE / FOODIE terms to AIM

FOODIE/INSPIRE term	type	AIM mapping	mapping_type	AIM module
foodie:Alert	class	fiware:Alert	equivalentClass	agriAlert
foodie:CropSpecies	class	saref4agri:Crop	equivalentClass	agriCrop
foodie:Plot	class	saref4agri:Parcel	equivalentClass	agriFeature
foodie:Product	class	fiware:AgriProductType	equivalentClass	agriProduct
foodie:PropertyTypeValue	class	saref:Property	equivalentClass	agriProperty
foodie:Treatment	class	fiware:AgriParcelOperation	equivalentClass	agriIntervention
inspire-af:FarmAnimalSpecies	class	saref4agri:Animal	equivalentClass	farmAnimal
inspire-af:Holding	class	saref4agri:Farm	equivalentClass	agriFeature
foodie:Alert	class	geosparql:Feature	subClassOf	agriAlert
foodie:CropSpecies	class	geosparql:Feature	subClassOf	agriCrop
foodie:FormOfTreatmentValue	class	skos:Concept	subClassOf	agriIntervention
foodie:Intervention	class	geosparql:Feature	subClassOf	agriIntervention
foodie:MachineType	class	sosa:Platform	subClassOf	agriSystem
foodie:ManagementZone	class	geosparql:Feature	subClassOf	agriFeature
foodie:OriginTypeValue	class	skos:Concept	subClassOf	agriFeature
foodie:Plot	class	geosparql:Feature	subClassOf	agriFeature
foodie:Product	class	geosparql:Feature	subClassOf	agriProduct
foodie:ProductKindValue	class	skos:Concept	subClassOf	agriProduct
foodie:ProductNutrients	class	geosparql:Feature	subClassOf	agriProduct
foodie:ProductPreparation	class	geosparql:Feature	subClassOf	agriProduct
foodie:PropertyTypeValue	class	skos:Concept	subClassOf	agriProperty
foodie:TractorType	class	sosa:Platform	subClassOf	agriSystem
foodie:Treatment	class	foodie:Intervention	subClassOf	agriIntervention
foodie:Treatment	class	geosparql:Feature	subClassOf	agriIntervention

foodie:TreatmentPlan	class	geosparql:Feature	subClassOf	agriIntervention
foodie:TreatmentPurposeValue	class	skos:Concept	subClassOf	agriIntervention
inspire-af:Holding	class	geosparql:Feature	subClassOf	agriFeature
inspire-af:Holding	class	inspire-act:ActivityComplex	subClassOf	agriFeature
inspire-af:Site	class	geosparql:Feature	subClassOf	agriFeature
foodie:crop	object_property	fiware:hasAgriCrop	equivalentProperty	agriFeature
foodie:interventionPlot	object_property	fiware:operationHasAgriParcel	equivalentProperty	agriIntervention
foodie:operator	object_property	fiware:hasOperator	equivalentProperty	agriIntervention
foodie:propertyType	object_property	saref:relatesToProperty	equivalentProperty	agriProperty
foodie:quantitativeProperty	object_property	sosa:hasResult	equivalentProperty	agriProperty
foodie:alertGeometry	object_property	geosparql:hasGeometry	subPropertyOf	agriAlert
foodie:containsPlot	object_property	saref4agri:contains	subPropertyOf	agriFeature
foodie:containsZone	object_property	saref4agri:contains	subPropertyOf	agriFeature
foodie:holdingPlot	object_property	saref4agri:isContainedIn	subPropertyOf	agriFeature
foodie:holdingSite	object_property	saref4agri:isContainedIn	subPropertyOf	agriFeature
foodie:holdingZone	object_property	saref4agri:isContainedIn	subPropertyOf	agriFeature
foodie:productionAmount	object_property	sosa:hasResult	subPropertyOf	agriProperty
inspire-af:contains	object_property	saref4agri:contains	subPropertyOf	agriFeature
foodie:alertPlot	object_property	n/a	reused	agriAlert
foodie:alertSpecies	object_property	n/a	reused	agriAlert
foodie:alertZone	object_property	n/a	reused	agriAlert
foodie:applicationWidth	object_property	n/a	reused	agriIntervention
foodie:areaDose	object_property	n/a	reused	agriIntervention
foodie:campaign	object_property	n/a	reused	agriIntervention
foodie:cropArea	object_property	n/a	reused	agriCrop
foodie:cropSpecies	object_property	n/a	reused	agriCrop
foodie:evidenceParty	object_property	n/a	reused	agriIntervention
foodie:flowAdjustment	object_property	n/a	reused	agriIntervention

foodie:formOfTreatment	object_property	n/a	reused	agriIntervention
foodie:ingredientAmount	object_property	n/a	reused	agriProduct
foodie:interventionGeometry	object_property	n/a	reused	agriIntervention
foodie:interventionZone	object_property	n/a	reused	agriIntervention
foodie:machine	object_property	n/a	reused	agriFeature
foodie:manufacturer	object_property	n/a	reused	agriProduct
foodie:maximumDose	object_property	n/a	reused	agriIntervention
foodie:minimumDose	object_property	n/a	reused	agriIntervention
foodie:motionSpeed	object_property	n/a	reused	agriIntervention
foodie:nutrient	object_property	n/a	reused	agriProduct
foodie:nutrientAmount	object_property	n/a	reused	agriProduct
foodie:nutrientProduct	object_property	n/a	reused	agriProduct
foodie:originType	object_property	n/a	reused	agriFeature
foodie:period	object_property	n/a	reused	agriIntervention
foodie:plan	object_property	n/a	reused	agriIntervention
foodie:planProduct	object_property	n/a	reused	agriIntervention
foodie:plotAlert	object_property	n/a	reused	agriAlert
foodie:pressure	object_property	n/a	reused	agriIntervention
foodie:productKind	object_property	n/a	reused	agriProduct
foodie:productQuantity	object_property	n/a	reused	agriProduct
foodie:production	object_property	n/a	reused	agriCrop
foodie:productionProperty	object_property	n/a	reused	agriProperty
foodie:quantity	object_property	n/a	reused	agriIntervention
foodie:safetyPeriod	object_property	n/a	reused	agriProduct
foodie:soilProperty	object_property	n/a	reused	agriProperty
foodie:solventQuantity	object_property	n/a	reused	agriProduct
foodie:speciesAlert	object_property	n/a	reused	agriAlert
foodie:supervisor	object_property	n/a	reused	agriIntervention

foodie:tractor	object_property	n/a	reused	agriFeature
foodie:treatmentProduct	object_property	n/a	reused	agriIntervention
foodie:zoneAlert	object_property	n/a	reused	agriAlert
inspire-af:activity	object_property	n/a	reused	agriFeature
inspire-af:includesAnimal	object_property	n/a	reused	farmAnimal
foodie:description	data_property	fiware:description	equivalentProperty	agriCommon
foodie:description	data_property	saref:hasDescription	equivalentProperty	agriCommon
foodie:livestockNumber	data_property	fiware:legalID	equivalentProperty	farmAnimal
foodie:livestockType	data_property	fiware:species	equivalentProperty	farmAnimal
foodie:productionDate	data_property	saref4agri:hasHarvestDate	equivalentProperty	agriCrop
foodie:status	data_property	fiware:status	equivalentProperty	agriIntervention
foodie:type	data_property	fiware:category	equivalentProperty	agriCommon
foodie:analysisDate	data_property	saref:hasTimestamp	subPropertyOf	agriProperty
foodie:ingredientName	data_property	fiware:name	subPropertyOf	agriProduct
foodie:nonQuantitativeProperty	data_property	saref:hasValue	subPropertyOf	agriProperty
foodie:nutrientName	data_property	fiware:name	subPropertyOf	agriProduct
foodie:productName	data_property	fiware:name	subPropertyOf	agriProduct
foodie:propertyName	data_property	fiware:name	subPropertyOf	agriProperty
foodie:alertDate	data_property	n/a	reused	agriAlert
foodie:code	data_property	n/a	reused	agriCommon
foodie:creationDateTime	data_property	n/a	reused	agriIntervention
foodie:family	data_property	n/a	reused	agriCrop
foodie:genus	data_property	n/a	reused	agriCrop
foodie:notes	data_property	n/a	reused	agriCommon
foodie:nutrientMeasure	data_property	n/a	reused	agriProduct
foodie:price	data_property	n/a	reused	agriCommon
foodie:productCode	data_property	n/a	reused	agriProduct
foodie:productSubType	data_property	n/a	reused	agriProduct

foodie:productType	data_property	n/a	reused	agriProduct
foodie:registerUrl	data_property	n/a	reused	agriProduct
foodie:registrationCode	data_property	n/a	reused	agriProduct
foodie:safetyInstructions	data_property	n/a	reused	agriProduct
foodie:species	data_property	n/a	reused	agriCrop
foodie:storageHandling	data_property	n/a	reused	agriProduct
foodie:treatmentDescription	data_property	n/a	reused	agriIntervention
foodie:treatmentPlanCode	data_property	n/a	reused	agriIntervention
foodie:treatmentPlanCreation	data_property	n/a	reused	agriIntervention
foodie:variety	data_property	n/a	reused	agriCrop

9.5 Semantic Mapping to AGROVOC

In this section, we talk about the AGROVOC *vocabulary,* which is a collection of component vocabularies related to the Agrifood, developed to support semantic representations and data modelling. It is arguably today the most complete multilingual controlled vocabulary for agriculture. This vocabulary contains over 40,000 concepts in a over 21 languages they cover a whole range of topics related to the agriculture sector such as food, nutrition, fishing, forestry, environment and other related sectors. The vocabulary looks like an RDF[5] using the SKOS[6] standard (i.e., the de-facto standard for sharing and linking knowledge organization systems as Linked Data[7]), while all concepts are identified by URL[8]. We have already presented theh AGROVOC vocabulary in more detail in D3.1; here we just summarize some key elements of it together with a summary of its inclusion (and mapping) into AIM.

Now, Figure 14 below presents the AGROVOC concept schema, based on SKOS model, where this schema is a top concept of all the others interpretation of AGROVOC model:

[5] https://www.w3.org/RDF/
[6] https://www.w3.org/2004/02/skos/
[7] https://www.w3.org/wiki/LinkedData
[8] https://www.w3.org/TR/url/

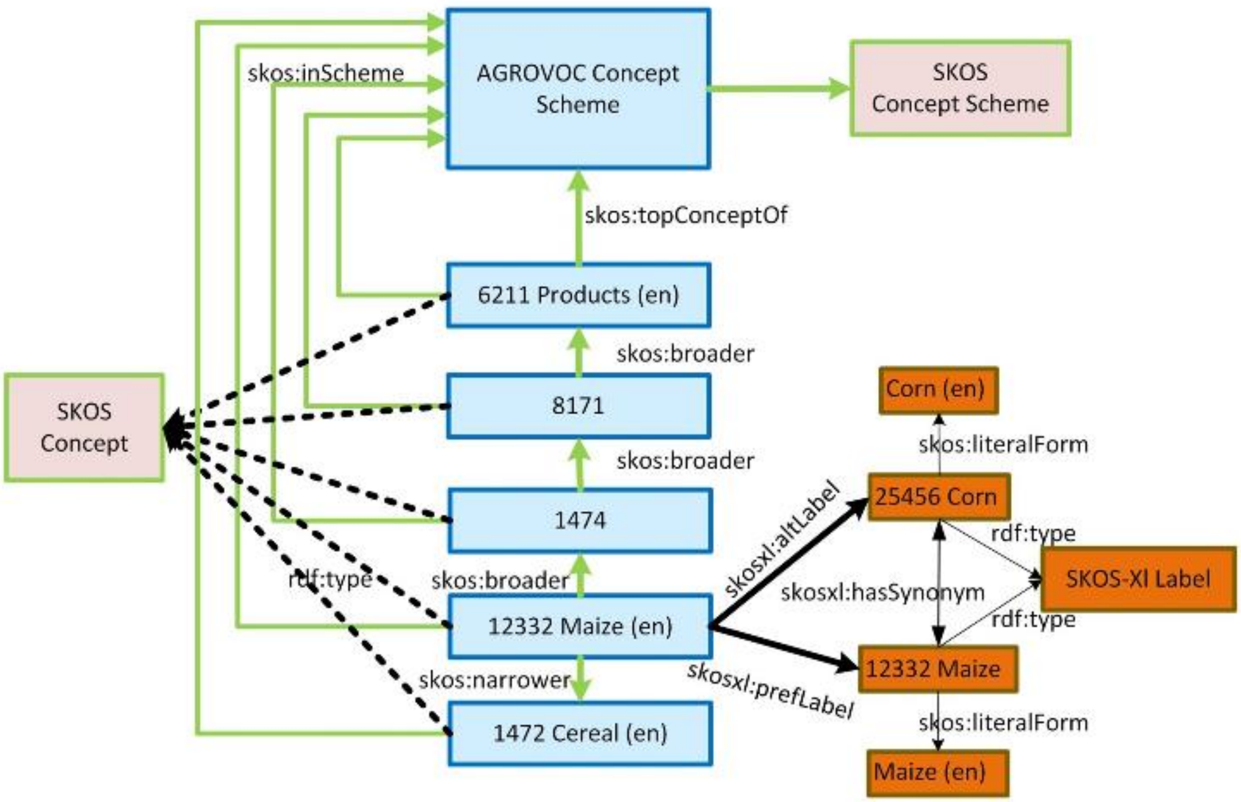

Figure 14. AGROVOC Concept Scheme [BoYa15]

With the introduction of Linking Open Data[9], the AGROVOC ontology has also been integrated with the others Linked Data vocabularies, effectively enabling semantic interoperability with other datasets and other ontologies and so on. This enables actors along the supply chain to use the same standards and vocabulary, or the integration of data within the supply chain with external data such as meteorological services and so on. Using this approach, every actor in the chain, from producer to consumer, can publish his data and link it to other entities of other vocabularies and ontologies. Consequently, AGROVOC provides these semantically explicit structured data according to a vocabulary/ontology reference easily readable by a machine as a sensor and by a variety of smart devices that are increasingly used throughout the Agrifood supply chain. Semantic technologies then allow the integration of data into information systems, being specifically designed for this purpose.

The amount of information in the AGROVOC vocabulary is enormous and, here, we deal precisely with this topic, that is to provide semantic interoperability between the DEMETER AIM model and the ontology model

[9] https://www.w3.org/egov/wiki/Linked_Open_Data

defined for AGROVOC. The primary objective is to implement an alignment between AIM by creating relationships, or better of the references between the entity defined in the AIM ontology and the relative references in the vocabulary: this allows the entity to connect to its semantic meaning. The link takes place using the Linked Data: both AIM as well as AGROVOC provide interfaces of this type. All entities URI[10] and properties to which specific agricultural concepts refer (e.g., temperature, humidity, pressure, etc.) can be aligned by means of the property derived from the RDF Scheme *<rdfs:isDefinedBy>*[11] with the corresponding URI of AGROVOC (e.g., the one relating to temperature http://aims.fao.org/aos/agrovoc/c_7657).

AIM can support this mapping as its meta-model can connect one relationship to another through the Linked Data mechanism. The AIM meta-model layer includes the concepts: *Entity*, *Relationship*, *Property* and *Value (NGSI-LD)*. The representation of the NGSI-LD meta-model reflects and extends the model *Entity-Relationship* (e.g., each relationship can be linked to another relationship, relationships can have properties) taking the form of property graph.

Now, the main integration with AGROVOC within AIM is done by re-using the agroVocConcept property from FIWARE data models that will have as value an AGROVOC concept URL. This integration allows data represented with AIM to be enriched with information about particular crops, plant products or pests, by connecting with the relevant AGROVOC concepts.

For instance, in AIM a farmer can describe their crops with all the relevant information, such as its status, when it was planted or where it was planted, and they can specify the particular crop species that was planted by referencing to the corresponding concept in AGROVOC, e.g., http://aims.fao.org/aos/agrovoc/c_7951 for common wheat. An excerpt of such declaration is provided in the table below.


```
{
      "@id": "urn:ngsi-ld:plot:72d9fb43-53f8-4ec8-a33c-fa931360259a",
      "@type": "Plot",
      "hasGeometry": {
        "@id": "urn:ngsi-ld:plot:geo:72d9fb43-53f8-4ec8-a33c-fa931360259y",
        "@type": "Polygon",
        "asWKT": "POLYGON (100 0, 101 0, 101 1, 100 1, 100 0)"
      },
      "area": 2012120,
      "description": "Spring wheat parcel",
      "category": "arable",
      "crop": {
        "@id": "urn:ngsi-ld:crop:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
        "@type": "Crop",
        "cropSpecies":{
          "@id": "urn:demeter:croptype:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
```


10 https://www.w3.org/wiki/URI
11 https://www.w3.org/2000/01/rdf-schema#

```
        "@type": "CropType",
        "name": "Wheat",
        "alternateName": "Triticum aestivum",
        "agroVocConcept": "http://aims.fao.org/aos/agrovoc/c_7951",
        "description": "Spring wheat"
      },
      "cropStatus": "seeded",
      "lastPlantedAt": "2016-08-23T10:18:16Z"
    }
  }
```


A further integration taking a more indirect approach can coexist, whereby AIM terms can be cross-referenced to AGROVOC equivalents. A sample of relevant AGROVOC concepts and potential mappings to DEMETER AIM classes are shown in Table 7.

Table 7. Mapping of AGROVOC OM Classes to AIM

AGROVOC Object Model Class	type	AIM mapping	mapping_type	AIM module
Animals	class	Saref4agri:AnimalGroup	equivalentClass	farmAnimal
Crop	class	FOODIE:CropType	equivalentClass	AgriCrop
Pests	class	FIWARE:AgriPest	equivalentClass	AgriPest
Soil	class	Saref4agri:Soil	equivalentClass	agriProperty

9.6 Semantic Mapping to EPPO

EPPO (European and Mediterranean Plant Protection Organization)[12] is an intergovernmental organization responsible for cooperation in plant health within the Euro-Mediterranean region. Founded in 1951 by 15 European countries, EPPO now has 52 members. Its objectives include:

- to protect plants health in agriculture, forestry and the uncultivated environment.
- to develop international strategies against the introduction and spread of pests which are a threat to cultivated and wild plants, in agricultural and natural ecosystems and protecting biodiversity
- to promote safe and effective pest control methods.

EPPO is a Regional Plant Protection Organization and participates in global discussions on plant health. EPPO is a standard-setting organization which has produced a large number of Standards in the areas of plant protection products and plant quarantine. These Standards constitute recommendations that are addressed to the National Plant Protection Organizations of EPPO member countries. Finally, EPPO promotes the exchange

[12] https://www.eppo.int/

of information between its member countries by maintaining information services and databases on plant pests, and by organizing many conferences and workshops.

One of the main outcomes of EPPO is its published Global Database[13]. The database is maintained by the EPPO Secretariat and aims to provide all pest-specific information that has been produced or collected by EPPO. The database contents are constantly being updated.

The database includes, among others, basic information for more than 90 000 species of interest to agriculture, forestry and plant protection: plants (cultivated and wild), animals and pests (including pathogens and invasive alien plants). For each species: scientific names, synonyms, common names in different languages, taxonomic position, and EPPO Codes are given.

Particularly useful for DEMETER is the collection and classification of plants and animal species, which can be used as reference for crop types, in similar way as the AGROVOC classification. The difference with AGROVOC is that EPPO typically provides much more detailed information of plants (see Figure 15 below). EPPO classification is widely used across different applications and organizations, including DEMETER's partners, and thus it was important to enable its usage from AIM.

[13] https://gd.eppo.int/

European Union
European Regional
Development Fund

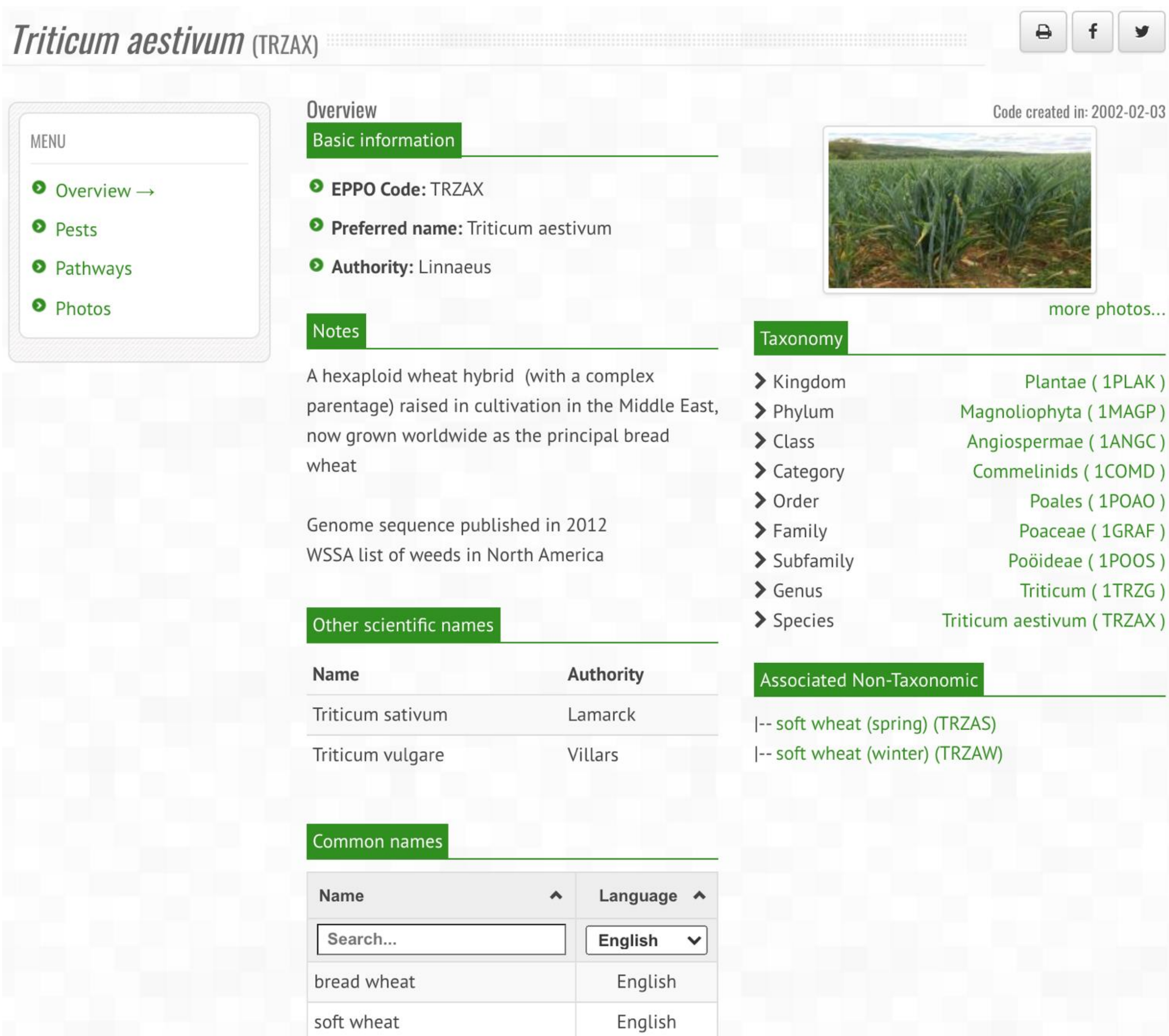

Figure 15. An example of the information available in EPPO for a specific crop

9.6.1 Semantic referencing to EPPO

Similar to AGROVOC, EPPO species are uniquely identified with a code, and they can be resolved on the Web via an URL that is composed by the EPPO service URL and the species code, and therefore they can be easily linked to other data following the Linked Data principles.

Hence, the main integration with EPPO within AIM is done by defining the eppoConcept property, in a similar way to the agroVocConcept property. As with AGROVOC, this integration allows data represented with AIM to be enriched with information about particular crops, animals or pests, by connecting to the relevant EPPO element.

The following tables shows how such an EPPO element can be used to enrich a crop description to include the particular crop species that was planted by referencing to the corresponding concept in EPPO, e.g., https://gd.eppo.int/taxon/TRZAX for rapeseed. This is a more complex/advanced description than the example for AGROVOC.


```
{
     {
        "@id": "urn:demeter:crop:72d9fb43-53f8-4ec8-a33c-fa931360259a",
        "@type": "Crop",
        "cropArea":{
               "@id": "urn:demeter:crop:geo:72d9fb43-53f8-4ec8-a33c-fa931360259a",
                       "@type": "Polygon",
                       "asWKT": "POLYGON (100 0, 101 0, 101 1, 100 1, 100 0)"
               },
        "cropSpecies": " urn:demeter:croptype:72d9fb43-53f8-4ec8-a33c-
fa931360259a",
        "cropStatus": "seeded",
        "lastPlantedAt": "2020-08-23T10:18:16Z",
        "validFrom":"30/1/2019",
        "validTo":"30/6/2019",
        "productionAmount": {
               "@type":"QuantityValue",
               "@id": "urn:demeter:productionAmount:72d9fb43-53f8-4ec8-a33c-
fa931360259a",
               "numericValue": 30,
               "unit": "http://qudt.org/vocab/unit/TONNE"
        }
     }
     {
        "@id": "urn:demeter:croptype:72d9fb43-53f8-4ec8-a33c-fa931360259a",
        "@type": "CropType",
        "code": "CropType2",
        "name": "Rapeseed" ,
        "family": "Brassicaceae",
        "description": "Rapeseed flowers are yellow and about 17 mm (0.67 in)
across. They grow to 100 cm in height with hairless, fleshy, pinnatifid and
glaucous lower leaves which are stalked whereas the upper leaves have no
petioles.",
        "species": "Brassica napus",
```

```
        "eppoConcept": "https://gd.eppo.int/taxon/TRZAX"
       }
}
```

9.7 *Semantic Mapping to Earth Observation standards*

In this subsection, we state the semantic mapping of AIM to Earth Observation (EO) data and standards. Now, as stated in D2.1, DEMETER needs to be able to represent both current earth observation (EO) data as well as historical EO data. AIM has been designed with several OGC best practices and standards that have been discussed in a more detail in D2.1 already. Overall, the interoperability with EO data has not changed significantly since then, so here, in this deliverable we restate some key pieces of information from D2.1, in the interest of completeness of information regarding the semantic interoperability of AIM with the well-known ontologies and models one of which is EO data models and standards.

Overall, the DEMETER AIM follows the OGC standard using GeoJSON, and even include a @context for GeoJSON, thus it is able to represent EO data. DEMETER defines the "data model" using the AIM model, and defines and adopts a profile of the OGC WCS that uses the NGSI-LD compatible encoding of the AIM model, with its basis in the OGC/W3C sosa:Observation. For example, the following snipet of code (following the AIM format) was already presented in D2.1:

```
{
      "@context":"http://schemas.opengis.net/os-geojson/1.0/os-geojson.jsonld",
      "type": "FeatureCollection",
      "id":
"https://services.terrascope.be/catalogue/products?collection=urn%3Aeop%3AVITO%3ATE
RRASCOPE_S2_FAPAR_V2",
      "features":
      {
            "type": "Feature",
            "id":
"urn:eop:VITO:TERRASCOPE_S2_FAPAR_V2:S2B_20191227T105349_31UFS_FAPAR_10M_V200",
            "geometry":
            {
                  "type": "Polygon",
                  "coordinates": "0 0"
            }
      },
      " properties":
      {
            "date": "2019-12-27T10:53:49Z",
            "updated": "2020-04-09T20:21:06Z",
            "available": "2020-04-16T15:00:32Z",
```

```
            "published": "2020-04-16T15:00:32Z",
            "status": "ARCHIVED",
            "parentIdentifier": "urn:eop:VITO:TERRASCOPE_S2_FAPAR_V2",
            "title": "S2B_20191227T105349_31UFS_FAPAR_10M_V200",
            "identifier":
"urn:eop:VITO:TERRASCOPE_S2_FAPAR_V2:S2B_20191227T105349_31UFS_FAPAR_10M_V200",
            "acquisitionInformation":
            {
                  "platform":
                  {
                        "platformShortName": "SENTINEL-2",
                        "platformSerialIdentifier": "S2B",
                  },
            },
      },
},
```

10 Revised Implementation of AIM and of Semantic Mappings

DEMETER AIM has been implemented following a layered and modular approach, reusing as much as possible existing ontologies and vocabularies, as described in the previous sections. In this section, we present the revised implementations for the different layers (parts) of AIM, implementation choices taken and the tools used during the implementation process. Links to the final results (i.e., the files implementing the AIM) are also provided.

10.1 Core meta-model

DEMETER AIM adopts and reuses the NGSI-LD meta-model, which provides a formal basis for representing "property graphs" using RDF/RDFS/OWL [OWL12]. It allows back and forth conversion between datasets that are based on the property graph model and linked data datasets, which rely on the RDF framework.

As described in the first release of AIM, which was presented in D2.1, we need to repeat that the meta-model defines the following entities (adapted from NGSI- LD [ETS18]):

- **Entity**: A DEMETER entity is defined as an NGSI-LD Entity, which is the informational representative of something that is supposed to exist in the real world, physically or conceptually. Any instance of such an entity shall be uniquely identified by a URI, and characterized by reference to one or more NGSI-LD Entity Type(s).
- **Property**: A DEMETER property is defined as an NGSI-LD property, which is a description instance that associates a main characteristic, which shall be a DEMETER Value, to either a DEMETER Entity, a DEMETER Relationship or another DEMETER Property. It shall include the special "hasValue" property to define its target value.
- **Value**: A DEMETER value is defined as an NGSI-LD Value that is either a JSON value (i.e. a string, a number, true or false, an object, an array), or a JSON-LD typed value (i.e. a string as the lexical form of the value together with a type, defined by an XSD base type or more generally a URI), or a JSON-LD structured value (i.e. a set, a list, a language-tagged string).
- **Relationship**: A DEMETER relationship is defined as an NGSI-LD Relationship that describes a directed link between a subject, which shall be either a DEMETER Entity, a DEMETER Property, or another DEMETER Relationship on one hand, and an object, which shall be a DEMETER Entity, on the other hand. It shall include the special "hasObject" property to define its target object.

The meta-model has been implemented primarily as a JSON-LD context in similar way as the NGSI-LD one (enabling the encoding of linked data in JSON). However, there are some differences:

- instead of defining cross-domain terms in the same context, as NGSI-LD does, the DEMETER meta-model is limited to the entities enabling the representation of “property graphs” described above.
- the entities use prefix “meta” (e.g., meta.Property) to avoid naming conflicts with other cross or

domain layer elements.

- the context does not define the mapping of "id" and "type" to JSON-LD keys "@id" and "@type" respectively, also to avoid conflicts in other layers.

DEMETER AIM meta-model JSON-LD implementation is available at: https://w3id.org/demeter/core-context.jsonld

Additionally, the meta-model has been implemented as an ontology, encoded in Turtle, in a similar way as the OWL-DL representation of the NGSI-LD provided in the documentation [ETS6]. However, as with the context, our implementation is limited to the entities enabling the representation of "property graphs" described above, without adding other cross-domain terms.

DEMETER AIM meta-model JSON-LD implementation is available at: https://w3id.org/demeter/core

10.2 Cross-Domain ontology

The conceptual changes stated in section 8.2 were technically integrated into the Cross-Domain ontology. Apart from adding the mentioned terms to the ontology, additional technical changes were required, which can be separated into the following categories:

The updated Cross-Domain Model is published as Turtle (TTL) file at: https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/crossdomainontology

The Cross-Domain SHACL validation file is located at: https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/crossdomainontology/-/tree/master/SHACL

The Domain-Specific Ontology GitLab repository features a description for end-users on how to use the above mentioned SHACL file for the validation of JSON-LD documents.

Added Imports:

The external ontologies/vocabularies that have been added to the cross-domain model since the initial release are listed in the following table.

Table 8. Added imports in the Cross-Domain layer of the second AIM release

Description	URL
SSN Extensions (ssn-ext)	http://www.w3.org/ns/ssn/ext/
Units and Quantities (qudt)	http://qudt.org/schema/qudt/ (downgraded from v.2.0)
NGSI-LD	https://uri.etsi.org/ngsi-ld/v1/ontology

Geographic Information - Observations and Measurements (iso19156_SF)	http://def.isotc211.org/iso19156/2011/SamplingFeature
Geographic Information - Observations and Measurements (iso19156_GFI)	http://def.isotc211.org/iso19156/2011/GeneralFeatureInstanc
Geographic Information - Observations and Measurements (iso19156_OB)	http://def.isotc211.org/iso19156/2011/Observation
Geographic Information Conceptual Schema Language (iso19103)	http://def.seegrid.csiro.au/isotc211/iso19103/2005/basic

Updated definition of schema elements:

The definition of Schema.org elements was changed, and two new elements were added, as indicated in the following table.

Table 9. Updated definition of schema elements in the Cross-Domain layer of the second AIM release

Type	Changes
https://schema.org/affiliation	changed rdf:type from owl:AnnotationProperty to rdf:Property
https://schema.org/domainIncludes	changed rdf:type from owl:AnnotationProperty to rdf:Property
https://schema.org/rangeIncludes	changed rdf:type from owl:AnnotationProperty to rdf:Property
https://schema.org/name	added
https://schema.org/telephone	added

Fixed link to AIM core (NGSI):

NGSI-LD core elements now refer to NGSI-LD directly, rather than referring to AIM core, as indicated in the following table.

Table 10. Fixed link to AIM core in the Cross-Domain layer of the second AIM release

Type	Changes
NGSI-LD Property	changed URL from

	https://w3id.org/demeter/core#Property to ngsi-ld:Property
NGSI-LD Relationship	changed URL from https://w3id.org/demeter/core#Relationship to ngsi-ld:Relationship
NGSI-LD Entity	changed URL from https://w3id.org/demeter/core#Entity to ngsi-ld:Entity

Furthermore, a SHACL file was added to facilitate the validation of instantiations of the Cross-Domain model.

The updated Cross-Domain Model is published as Turtle (TTL) file at: https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/crossdomainontology

The Cross-Domain SHACL validation file is located at:

https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/crossdomainontology/-/tree/master/SHACL

The Domain-Specific Ontology GitLab repository[14] features a description for end-users on how to use the above mentioned SHACL file for the validation of JSON-LD documents.

10.3 Domain-specific ontologies

The changes captured in section 8.3 redresses some structural weakenings in the implementation of the domain-specific modules. More specifically, there have been found several terms that were used or even defined across multiple modules, this resulting into sub-optimal definition of the whole AIM, though this causes no functional failure.

These structural infelicities are identified and corrected and the majority of the aforementioned terms have been moved to the upper (cross-domain layer) and the rest are defined only in the relevant domain-specific ontology. Apart from the more meaningful changes described above, every ontology changed URL references to W3C, in order to have a more stable link.

The ontologies altered in this release are:

- agriCommon.ttl
- agriFeature.ttl
- agriProperty.ttl

[14] https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies

- agriSystem.ttl

10.4 Pilot-specific ontologies

As explained in Section 8.4, pilot-specific layer comes as a result of the increased need of the pilots to extend the AIM with ontologies and terms that do not exist in any well-known ontology. Initially, some extensions of the domain-specific ontologies were created and then the reasons explained in the respective section for the design led us to create this additional layer of DEMETER AIM.

These ontologies are implemented following the same principles of the domain-specific layer. However, they differ significantly since there is slight re-usage of known terms and they mostly concern some alignments to classes that are defined previously. Naturally, the modules are mostly flat and hold few links to the layers above or other ontologies. The way that this link is implemented is through importing the most relevant domain-specific ontology, since these modules normally fall under some agricultural domain. In case there is no such agri-food sector implemented in AIM, there is direct import of the cross-domain ontology.

The pilot-specific layer covers specialized needs of the pilots, as captured in the questionnaires described in Chapter 6 and the current implemented modules are:

- fieldOperations.ttl
- kpiIndicator.ttl
- livestockFeature.ttl
- nutrientMonitor.ttl
- poultryFeeding.ttl
- stressRecognition.ttl
- transportCondition.ttl

10.5 Metadata Schema

For the metadata schema implementation, an application profile of the IDS Information Model is created and adapted to suit the particular needs of DEMETER as described in section 7. The original IDS Information Model is hosted on GitHub at https://github.com/International-Data-Spaces-Association/InformationModel/, the DEMETER specific application profile at https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/metadataschema.

More detailed documentation on the implementation can be found in the respective DEMETER GitLab repository. The DEMETER application profile of the IDS Information Model will be maintained by DEMETER. By its construction as an application profile, we benefit from improvements and new features of the original IDS Information Model, which is actively maintained. Here it should be ensured that new IDS features do not break with DEMETER requirements and/or the DEMETER application profile will be adapted accordingly.

The implementation and documentation of the original IDS infomodel is available at: https://w3id.org/idsa/core

10.6 AIM publication and profiling

The cross-domain ontology of AIM is a graph closure (aggregation) of a set of virtual profiles of the standards used by AIM. The AIM model publication process analyses the elements in use and creates actual profile model declarations that allow the cross-domain ontology usage to be explicitly identified and create the opportunity for interoperability with other domain models that use similar subsets.

Use of an element from ontology B in an ontology B, or an owl:imports statement does not declare the intent that data instances are interoperable with another – that is a function of the details of class declarations in ontologies A and B. Such details require significant knowledge of OWL constructs, and may even require importing multiple other ontologies and performing reasoning on these to determine relationships.

Profile descriptions provide the following additional "value" – i.e., accessible information:

- Creates an explicit link between an AIM model and the cross-domain models it profiles (and hence data is interoperable with)

The following statement make it explicit that instance data conforming to aim:agriAlert model also conforms to FIWARE and FOODIE models:

```
aim:agriAlert
      prof:isProfileOf
            <https://uri.fiware.org/ns/data-models>,
            <https://w3id.org/demeter/agri/proxy/foodie>;
```

- Details the exact relationships between the AIM model and the profiled standards. These relationships may be:
    - requirements that an optional property is mandatory in the profile
    - specification of sub-types of generalized classes for use in the profile domain
    - alignments specifying equivalence.

Eg. a simple statement defining equivalence within the AIM model between two similar standards:

```
Foodie:Alert owl:equivalentClass fiware:Alert
```

This is an effective restriction on the usage of both classes that AIM implementation must be compatible with both profiled data models, and hence can interoperate with both FOODIE and FIWARE domains.

The key pattern prevalent in profiling of standards is the creation of separate file with the desired subset of profiled ontology mixed with additional assertions. This leads to a challenge in that the presence of these duplicate descriptions in a particular file is the statement of intent regarding profile definition. This presence is not directly visible once the contents of the file are read into a semantic data store. Various tricks, such as creating a context (or graph) name for the file and always adding this into queries are possible - however this is not a standardized behaviour and creates a far more complex problem for users of a data model to find out what constraints a profile introduces

```
SELECT ?profile ?aClass ?standard ?p ?o


WHERE {

      GRAPH   ?profie # value of which must be identified and set when
reading a file

            { ?aClass a owl:Class .

              ?a Class rdfs:isDefinedBy ?standard. #this needs to be
standardised too

              ?aClass ?p ?o

            MINUS { GRAPH{?standard [?aClass ?p2 ?o2}} # which
probably wont work for blank nodes anyway

            }

      # other things we care about

      }
```

The purposes of formalizing a profile can add explicit statements about these relationships. The standardization of these descriptions is a work in progress.

- Links to other resources that capture implicit information in the profile that is not otherwise semantically declared, and cannot be discovered without adding additional information to the system.
- Generation of implementation resources such as "frames" – or "schema" describing the basic structure of classes and properties compliant with the profile. These may take many forms, such as:
    - SHACL shapes – semantic expression of frames
    - JSON-schema – JSON encoding constraints
    - RDF- Datacube qb:DataStructureDefinition
    - XML schema
    - Etc.

These derived profiles have been automatically generated and published as part of the AIM model publication process.

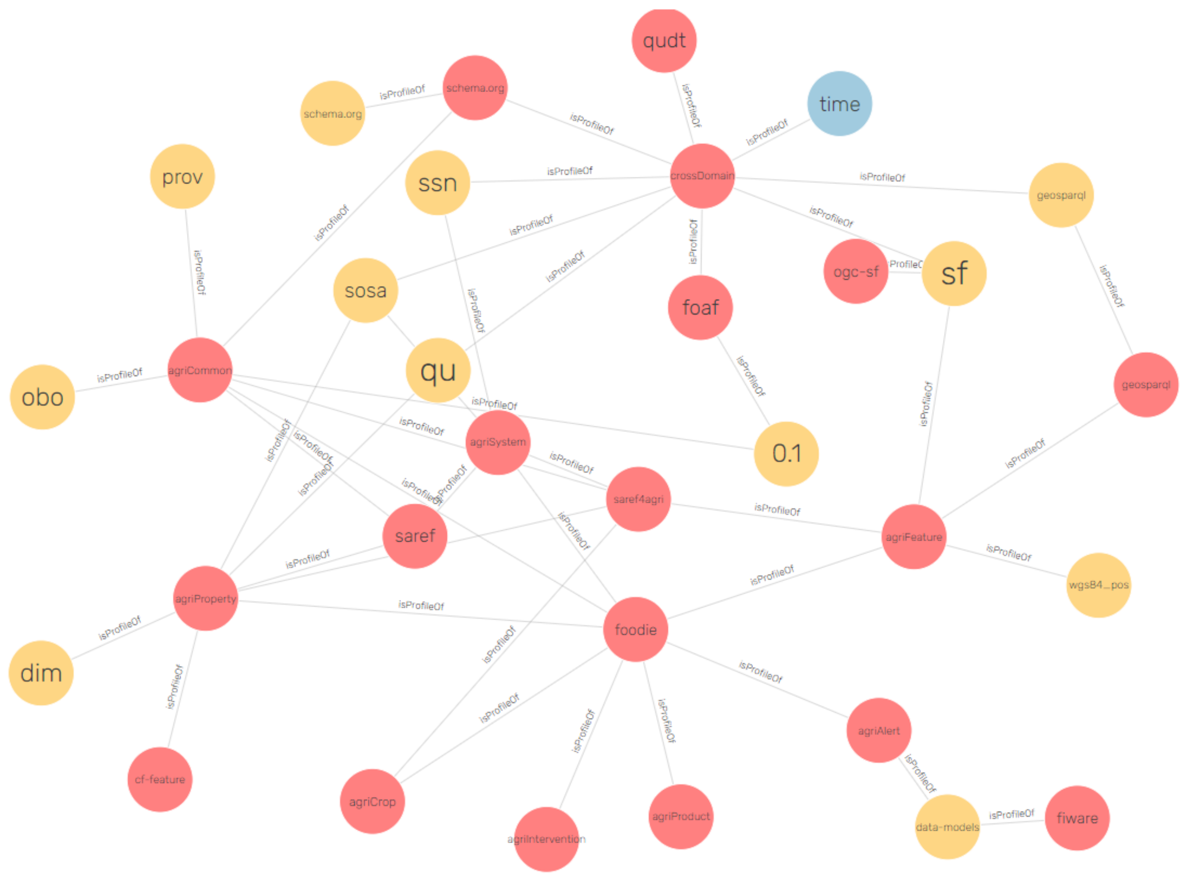

Figure 16. Preliminary view of inferred profile relationships in AIM

Finally, the main update to AIM is to characterize the use of the SOSA ontology. This has been characterized through publication of a set of formal profiles for SOSA for use in different sub-domains.

10.6.1 Emergent profiles

The AIM model sets out interoperability requirements through reuse of externally defined models. Any application of an interoperability specification must also make implementation decisions about the content to be expressed using these specifications. The interoperability of the content is also a key element for long-term scalability. Using semantic model standards we can, for example, say that the range of property is a skos:Concept, which implies a skos:ConceptScheme – in other words that the value comes from some controlled list of terms, but the model itself does not define the source of these terms.

The DEMETER project has a suite of pilots. – Due to the needs of each of these pilots, another layer of profiling can emerge that defines the use of particular vocabularies for different application domains. These may be unique to a pilot within the DEMETER project. However, from the perspective of the longer-term goal describing these constraints as a suite of implementation profiles of AIM provides an opportunity to create a richer ecosystem of interoperable applications in future. Of more interest however is the potential to describe these profiles against the reusable cross-domain models profiled in the AIM cross-domain model.

The proposed way to describe content usage restrictions is through the RDF Datacube component of the AIM Cross-domain model.

For an example, a restriction on use of the SOSA model (W3C/OGC Sensor Observation Sampling & Actuation Ontology) might be on the use of a particular taxonomy of observed properties:

```
<urn:demeter:observation-20180101>

  a sosa:ObservationCollection ;

  sosa:hasFeatureOfInterest <urn:demeter:plot:72d9fb43-53f8-4ec8-
a33c-fa931360259a>;

  sosa:hasMember <urn:demeter:observation/20180101/q10>,

<urn:demeter:observation/20180101/q50>,

<urn:demeter:observation/20180101/q90> ;

  sosa:observedProperty <http://purl.oclc.org/NET/ssnx/cf/cf-
property#normalized_difference_vegetation_index> ;

  sosa:resultTime "2018-01-01T12:36:12Z"^^xsd:dateTime .
```

where:

```
<http://purl.oclc.org/NET/ssnx/cf/cf-property> a skos:ConceptScheme
;

      rdfs:comment "Derived from the ontology" .

http://purl.oclc.org/NET/ssnx/cf/cf-
property#normalized_difference_vegetation_index a skos:Concept ;

      a sosa:ObservableProperty
```

This profile requirement may be described using an RDF Datacube description; (fragment only)

```
sosa:observedProperty a qb:CodedProperty;

qb:codelist<http://purl.oclc.org/NET/ssnx/cf/cf-property> ; # all
members must also be members of the CF-Property

rdfs:range sosa:ObservableProperty
```

All these relationships and declarations of object types can be derived from the combination of AIM, sample data and the vocabularies in use. This is a process requiring a great deal of expertise, and curation of the set of declarations needed to semantically link these concepts.

Formalised profiles can encapsulate these details and make them directly accessible to implementers and users of collections of data generated by implementing systems. These profiles are *emergent* in that they will be used to describe the usage of AIM *as implemented* in pilots, and form examples of how AIM may be used in future to support highly scalable and interoperable ecosystems of implementations.

11 Usage of AIM across the pilots

AIM, including the support for semantic interoperability it provides, is part of the core DEMETER enablers. Therefore, its usage is mandatory across all the DEMETER pilots. Based on a questionnaire circulated during a developer workshop conducted 29-31/3/2021, the usage of AIM across the pilots is as follows:

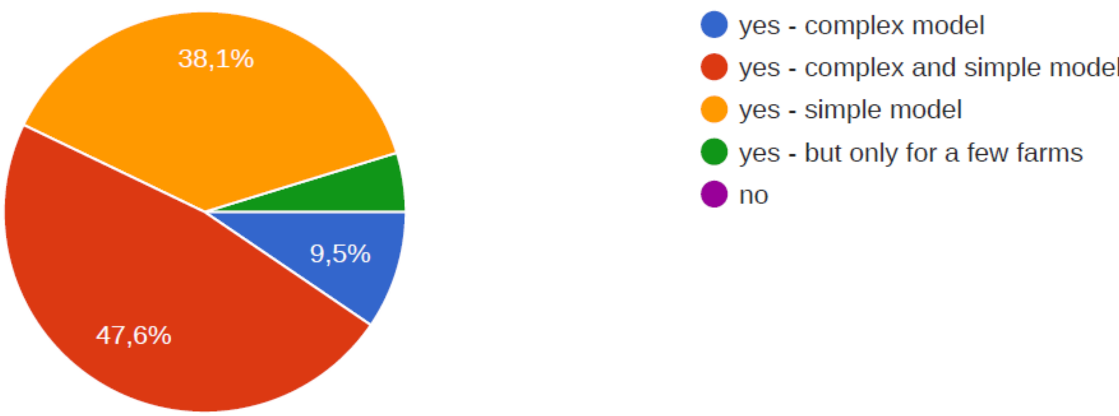

Figure 17. Summary of responses of pilot developers to the survey question: "*Are you planning to use AIM to describe farms, concepts and data of your pilot?*"

In the rest of this section, the usage of the AIM across the pilots is further detailed, the AIM-related support required by the pilots iselaborated upon, and the further AIM extensions originating from the pilots are discussed, while AIM usage guidelines for pilot developers are also provided.

11.1 AIM usage guidelines for pilot developers [ICCS]

DEMETER provides guidelines on how to use AIM[15], including instructions on how to find and identify relevant terms and how to create AIM-based JSON-LD content, a discussion of key terms and examples, as well as, instructions to validate the generated content.

11.1.1 Finding and identifying relevant concepts

Users and developers have different options to find terms in AIM and to navigate its structure:

- For data modelers, the best option is to load the whole ontology in an ontology editor like Protege[16] using the main AIM URL: https://w3id.org/demeter/agri. Once loaded, start searching for terms, or

[15] https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies
[16] https://protege.stanford.edu/

navigating the structure of classes, object properties or data properties.

- For developers/advanced users, all URLs of modules of AIM domain layer (e.g., https://w3id.org/demeter/agri, https://w3id.org/demeter/agri/agriCrop) take you by default to OGC definiton server (unless using content negotiation for TURTLE or RDF/XML that gives you directly the source). From the OGC server you can search for terms in all those modules via http://defs-dev.opengis.net/demeter/search. Also from OGC server many related resources are accessible via content-negotiation-by-profile, using the profile "alt" (linked from HTML views) - e.g., https://w3id.org/demeter/agri/agriCrop?_profile=alt. These include JSON-LD schema, HTML documentation views, different RDF serializations and other resources.
- For typical users: AIM is available via agroPortal: http://agroportal.lirmm.fr/ontologies/DEMETER-AIM. From there you can browse through the taxonomies of classes or properties, or display notes or mappings if any (e.g., http://agroportal.lirmm.fr/ontologies/DEMETER-AIM?p=classes. Additionally, you can use the recommender[17] and the annotator[18] functionalities of the portal. The former gives recommendations for the most relevant ontologies based on an excerpt from a text or a list of keywords, optionally highlighting the annotations in the result. For example, giving the recommender a typical DEMETER use case description: “I am a farmer, and I have a farm that has a parcel with a maize crop and a second parcel with a wheat crop”, it returns AIM at the top as the most relevant ontology (see Figure 18 below), and highlights the words in the text with corresponding elements in the ontology (which can be clicked to open their description). Similarly, we can use the annotator to get the AIM terms corresponding to the identified words in the text (see Figure 19 below).

[17] http://agroportal.lirmm.fr/recommender
[18] http://agroportal.lirmm.fr/annotator

Figure 18. Agroportal ontology recommender functionality

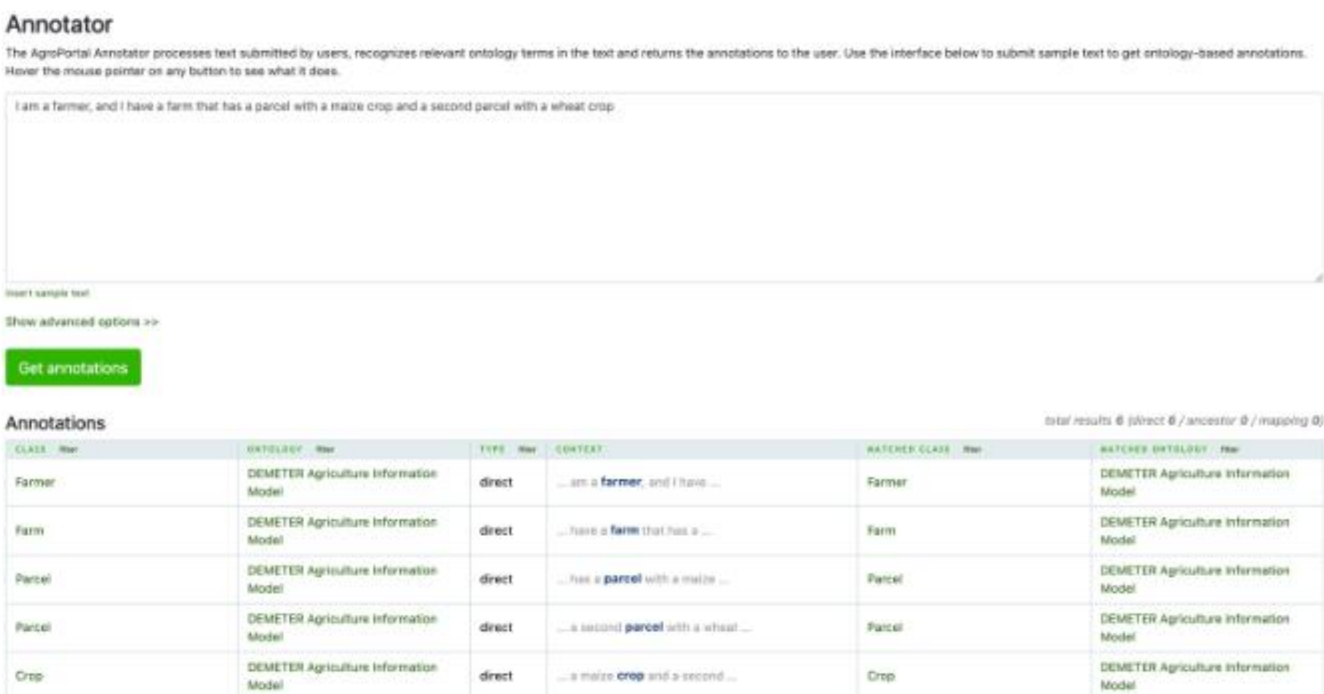

Figure 19. Agroportal annotator functionality

- For developers, Agroportal features are also available via API:
  - search *for terms in AIM*[19] *(e.g., Plot)* http://data.agroportal.lirmm.fr/search?ontologies=http://data.agroportal.lirmm.fr/ontologies/DEMETER-AIM&q=plot&apikey=)
  - search *for properties in AIM*[20] *(e.g., hasAgriParcel)* http://data.agroportal.lirmm.fr/property_search?ontologies=http://data.agroportal.lirmm.fr/ontologies/DEMETER-AIM&q=hasAgriParcel&apikey=)
  - *get* annotations *(terms from AIM) given an input text (e.g., I am a farmer, I have a farm that has a parcel with a maize crop and a second parcel with a wheat crop)*[21] *(e.g., http://services.agroportal.lirmm.fr/annotator/?text=I+am+farmer%2C+I+have+a+farm+that+has+a+parcel+with+a+maize+crop+and+a+second+parcel+with+a+wheat+crop&ontologies=AGROVOC,DEMETER-AIM&apikey=)*

11.1.2 How to create AIM-based JSON-LD content

Basic JSON-LD terms

JSON-LD is designed around the concept of a "context" to provide mappings from JSON to a shared/common model, allowing applications to use shortcuts to terms to communicate with one another more efficiently, but without losing accuracy.

- The context links terms in a JSON document to elements in an ontology or vocabulary, i.e., AIM in the case of DEMETER.
- Your json-ld content will have "@graph" entry where you can define multiple objects in your data payload
- Each json-ld objects should have
  - "@id" key to uniquely identify node objects that are being described in the document using IRIs (urn:demeter:ag:88acd214-f633-4db7-9560-0ca69abc1a4a, or https://myorganization.org/objects/88acd214-f633). Note this is an identifier of the object describing a data element (e.g., a plot), which can be an internal or externally resolvable, and it is different than the identifier of the data element described in it (e.g., id of the plot) at the application level.
  - "@type" key to set the type of a node or the datatype of a typed value (from the context)

[19] https://tinyurl.com/2xd4hmyz
[20] https://tinyurl.com/2hkepvre
[21] https://tinyurl.com/dwsth5da

AIM-based JSON-LD content

In order to generate AIM-based JSON-LD content, you need to define the @context in your JSON document, and reference AIM context(s) from there. In general, the simplest method is to specify just the main AIM context as below, which includes all terms in AIM (from the all the layer, except pilot-specific extensions).

```
{
  "@context": [
          "https://w3id.org/demeter/agri-context.jsonld"
   ],
  "@graph": [

       <<data goes here>>

   ]
}
```

Alternatively, you can use individual contexts, e.g., to use only few modules of AIM. Note that referencing the main context is the same as referencing all individual modules (including cross-domain and domain layer) as shown below.

```
{
  "@context": [
        "https://w3id.org/demeter/agri/crossDomain-context.jsonld",
        "https://w3id.org/demeter/agri/agriFeature-context.jsonld",
        "https://w3id.org/demeter/agri/agriCrop-context.jsonld",
        "https://w3id.org/demeter/agri/agriCommon-context.jsonld",
        "https://w3id.org/demeter/agri/agriIntervention-context.jsonld",
        "https://w3id.org/demeter/agri/agriAlert-context.jsonld",
        "https://w3id.org/demeter/agri/agriProduct-context.jsonld",
        "https://w3id.org/demeter/agri/agriProperty-context.jsonld",
        "https://w3id.org/demeter/agri/agriSystem-context.jsonld",
        "https://w3id.org/demeter/agri/agriPest-context.jsonld",
        "https://w3id.org/demeter/agri/farmAnimal-context.jsonld"
   ],
  "@graph": [

       <<data goes here>>

   ]
}
```

Additionally, you should include any pilot specific contexts relevant for your pilot/application, e.g.,

```
{
  "@context": [
          "https://w3id.org/demeter/agri-context.jsonld",
           "https://w3id.org/demeter/agri/ext/poultryFeeding-context.jsonld"
   ],
  "@graph": [

       <<pilot specific data goes here>>

   ]
}
```

It should be noted that there are two ways to reference JSON-LD contexts:

- using the ontology URI+"-context.jsonld" e.g., https://w3id.org/demeter/agri/agriFeature-context.jsonld (the recommended and standard form, used in examples);
- using the ontology URI+profile parameter (thanks to OGC definition server), e.g., https://w3id.org/demeter/agri/agriFeature?_profile=jsoncontext. Nevertheless, this approach still requires some updates to handle issues with entities with duplicated names. So, in the meantime, please use first approach.

11.1.3 AIM key terms discussion

As mentioned in Section 8, AIM builds upon different standard and well-known vocabularies. In some cases, though, these models/vocabularies are able to represent the same concepts or properties, but using different terms (e.g., Plot in FOODIE same as Parcel in Saref4Agri). Hence, two JSON-LD contents that are AIM compatible could use different terms to represent the same information. This should not be a problem since we have implemented the semantic mappings between those terms in different vocabularies (see Section 9). However, this would require the applications to load and understand all those mappings to realize that both contents represent the same information. So, to facilitate the exchange of data between DEMETER components, we provide some reference terms, at least to address the most common scenarios.

Data element identifier (application level)

Use attribute identifier to associate the identifier of the object in the application, instead of defining your own identifier, e.g, tractorId, animalId, etc.

Farm related features

Farm features can be described at different level of granularity, where a simple farm would use typically only 2 levels, a more complex farm may need three or four.

Level	AIM term (reference)	Saref4Agri term	FIWARE term	FOODIE term	ADAPT term
1	Farm	Farm	AgriFarm	Holding	Farm
2	Site	-	-	Site	-
3	Plot	Parcel	AgriParcel	Plot	Field
4	ManagementZone	-	-	ManagementZone	-

Crops

Most DEMETER applications need to define/describe crops.

Level	AIM term (reference)	Saref4Agri term	FIWARE term	FOODIE term	ADAPT term
1	Crop	Crop	AgriCrop	CropSpecies	CropZone
2	CropType (object defining the particular crop type/species)	taxonomic_rank (taxrank vocabulary)	-	CropType	Crop
3	cropSpecies (property associating Crop to CropType)	has_rank (taxrank vocabulary)	-	cropSpecies	-
4	agroVocConcept	-	agroVocConcept	-	-
5	eppoConcept	-	-	-	-

Geospatial properties

To represent the geographical area associated to the land, the following properties are used:

- Reference AIM property: “location” to associate countries/regions/municipalities where the land is located or a Point (with lat/long/alt)
- Reference AIM property: “hasGeometry” to associate the geospatial information (e.g., polygon/multipolygon)

Time series

To represent time series (e.g., multiple observations/measurements over a period of time), we follow the SOSA/SSN model and approach. This means that we model each of those observations as a SOSA:Observation, that has associated:

- feature of interest (e.g., Crop, Field, Tractor) (via sosa:hasFeatureOfInterest)

- the observed property (e.g., temperature, density, position) (via sosa:observedProperty)
- the result of the observation (which has a numerical value and a unit) (via sosa:hasResult) or a simple value (via sosa:hasSimpleResult).
- the time of the observation (via sosa:resultTime) and
- potentially the sensor used to make the observation (via sosa:madeBySensor).

Statistical data

To represent statistical data (e.g., agri indicators), we follow the RDF data cube model and approach. A statistical data set comprises a collection of observations made at some points across some logical space. The collection can be characterized by

- a set of dimensions that define what the observation applies to (e.g., time, area, gender) along with
- metadata describing what has been measured (e.g., economic activity, population),
- how it was measured and
- how the observations are expressed (e.g., units, multipliers, status).

We can think of the statistical data set as a multi-dimensional space, or hyper-cube, indexed by those dimensions. This space is commonly referred to as a cube for short; though the name shouldn't be taken literally, it is not meant to imply that there are exactly three dimensions (there can be more or fewer) nor that all the dimensions are somehow similar in size. A cube is organized according to:

- Dimension components that serve to identify the observations. A set of values for all the dimension components is sufficient to identify a single observation (e.g., the time to which the observation applies, or a geographic region which the observation covers).
- The measure components represent the phenomenon being observed.
- The attribute components allow to qualify and interpret the observed value(s). They enable specification of the units of measure, any scaling factors and metadata such as the status of the observation (e.g., estimated, provisional).

11.1.4 Examples

We provide a number of examples of AIM compliant JSON-LD documents showcasing many of the main elements in the model, and especially relevant to multiple DEMETER pilots. The examples are provided both as JSON-LD documents and as Turtle documents. These examples include, among others:

- A simple farm description, including plots and crops
- A complex farm description, including sites, and multiple plots and crops with more details.
- Multiple time series examples showcasing how to represent different data using SOSA/SSN approach,

many of these examples are from DEMETER pilots

- Pilot specific data that extend AIM domain layer modules
- Counter examples showcasing incorrect use of AIM

The table below illustrates the simple farm example, describing a farm with two plots, and each plot with one crop, whereas the table after that presents a time series example describing three observations (aggregated in a collection of observations) made over a Plot.


```
{
  "@context": [
              "https://w3id.org/demeter/agri-context.jsonld"
   ],
  "@id": "urn:ngsi-ld:farm:72d9fb43-53f8-4ec8-a33c-fa931360259a",
  "@type": "Farm",
  "name": "Wheat farm",
  "description": "A farm producing wheat",
  "hasGeometry": {
    "@id": "urn:ngsi-ld:AgriFarm:geo:72d9fb43-53f8-4ec8-a33c-fa931360259x",
    "@type": "Point",
    "asWKT": "POINT(11.3 44.12)"
  },
  "containsPlot":[
    {
      "@id": "urn:ngsi-ld:plot:72d9fb43-53f8-4ec8-a33c-fa931360259a",
      "@type": "Plot",
      "hasGeometry": {
        "@id": "urn:ngsi-ld:plot:geo:72d9fb43-53f8-4ec8-a33c-fa931360259y",
        "@type": "Polygon",
        "asWKT": "POLYGON (100 0, 101 0, 101 1, 100 1, 100 0)"
      },
      "area": 2012120,
      "description": "Spring wheat parcel",
      "category": "arable",
      "crop": {
        "@id": "urn:ngsi-ld:crop:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
        "@type": "Crop",
        "cropSpecies":{
          "@id": "urn:demeter:croptype:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
                "@type": "CropType",
          "name": "Wheat",
          "alternateName": "Triticum aestivum",
          "agroVocConcept": "http://aims.fao.org/aos/agrovoc/c_7951",
          "description": "Spring wheat"
        },
        "cropStatus": "seeded",
```

```
        "lastPlantedAt": "2016-08-23T10:18:16Z"
      }
    },
    {
      "@id": "urn:ngsi-ld:plot:72d9fb43-53f8-4ec8-a33c-fa931360259b",
      "@type": "Plot",
      "hasGeometry": {
        "@id": "urn:ngsi-ld:AgriParcel:geo:72d9fb43-53f8-4ec8-a33c-fa931360259z",
        "@type": "Polygon",
        "asWKT": "POLYGON (100 0, 101 0, 101 1, 100 1, 100 1)"
      },
      "area": 200,
      "description": "Spring barley parcel",
      "category": "arable",
      "crop": {
        "@id": "urn:ngsi-ld:crop:df72dc57-1eb9-42a3-88a9-8647ecc954b5",
        "@type": "Crop",
        "cropSpecies":{
          "@id": "urn:demeter:croptype:df72dc57-1eb9-42a3-88a9-8647ecc954b5",
                     "@type": "CropType",
          "name": "Barley",
          "alternateName": "Ordeum",
          "agroVocConcept": "http://aims.fao.org/aos/agrovoc/c_7952",
          "description": "Spring barley"
        },
        "cropStatus": "seeded",
        "lastPlantedAt": "2016-08-23T10:18:16Z"
      }
    }
  ]
}
```



```
{
  "@context": [
    "https://w3id.org/demeter/agri-context.jsonld",
          {
      "qudt-unit": "http://qudt.org/vocab/unit/"
    }
   ],
   "@graph": [
        {
      "@id": "urn:demeter:plot:72d9fb43-53f8-4ec8-a33c-fa931360259a",
      "@type": "Plot",
      "hasGeometry": {
```

```
        "@id": "urn:demeter:plot:geo:72d9fb43-53f8-4ec8-a33c-fa931360259y",
        "@type": "Polygon",
        "asWKT": "POLYGON (100 0, 101 0, 101 1, 100 1, 100 0)"
      },
      "area": 2012120,
      "description": "Spring wheat plot",
      "category": "arable",
      "crop": {
        "@id": "urn:demeter:crop:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
        "@type": "Crop",
        "cropSpecies": "urn:demeter:croptype:df72dc57-1eb9-42a3-88a9-
8647ecc954b4",
        "cropStatus": "seeded",
        "lastPlantedAt": "2016-08-23T10:18:16Z"
      }
    },
    {
      "@id": "urn:demeter:croptype:df72dc57-1eb9-42a3-88a9-8647ecc954b4",
      "@type": "CropType",
      "name": "Wheat",
      "alternateName": "Triticum aestivum",
      "agroVocConcept": "http://aims.fao.org/aos/agrovoc/c_7951",
      "eppoConcept": "https://gd.eppo.int/taxon/TRZAX",
      "description": "Spring wheat"
    },
    {
      "@id": "urn:demeter:observation-20180101",
      "@type": "ObservationCollection",
      "observedProperty": "http://purl.oclc.org/NET/ssnx/cf/cf-
property#normalized_difference_vegetation_index",
      "hasFeatureOfInterest": "urn:demeter:plot:72d9fb43-53f8-4ec8-a33c-
fa931360259a",
      "madeBySensor": "sensor/35-207306-844818-0/BMP282",
      "resultTime": "2018-01-01T12:36:12Z" ,
      "hasMember": ["urn:demeter:observation/20180101/q10",
"urn:demeter:observation/20180101/q50","urn:demeter:observation/20180101/q90"]
    },
    {
      "@id": "urn:demeter:observation/20180101/q10",
      "@type": "Observation",
      "identifier": "q10",
      "hasSimpleResult": "0.27121272683143616"
     },
     {
      "@id": "urn:demeter:observation/20180101/q50",
```

```
      "@type": "Observation",
      "identifier": "q50",
      "hasResult": {
        "@id": "urn:demeter:observation/20180101/q50/result",
        "@type": "QuantityValue",
        "numericValue": "0.3173256516456604",
        "unit": "qudt-unit:UNITLESS"
     }
     },
     {
      "@id": "urn:demeter:observation/20180101/q90",
      "@type": "Observation",
      "identifier": "q90",
      "hasResult": {
        "@id": "urn:demeter:observation/20180101/q90/result",
        "@type": "QuantityValue",
        "numericValue": "0.38018566370010376",
        "unit": "qudt-unit:UNITLESS"
      }
     }
  ]
}
```

11.1.5 How to validate your JSON-LD content is AIM-compliant

There are two levels of validation that can be carried out. The first, basic level, is to validate that all the terms used in your JSON-LD are recognized AIM terms, from any of the first three layers or from the extensions used. A simple way to validate this is using the JSON-LD playground[22]. The most important aspect of this validation is not only to make sure that the JSON-LD is syntactically correct, but also that all elements in the input appear in the output as valid AIM elements. Note that the latest AIM context includes a default namespace, so if a term is not recognized as AIM term, it will appear as "https://w3id.org/demeter/default-context/{term}". This means that this term should be updated to use an AIM term or request an extension for it. The figure below shows the simple farm example in the playground, and you can also see it online[23].

[22] https://json-ld.org/playground/
[23] https://tinyurl.com/azfvt4x4

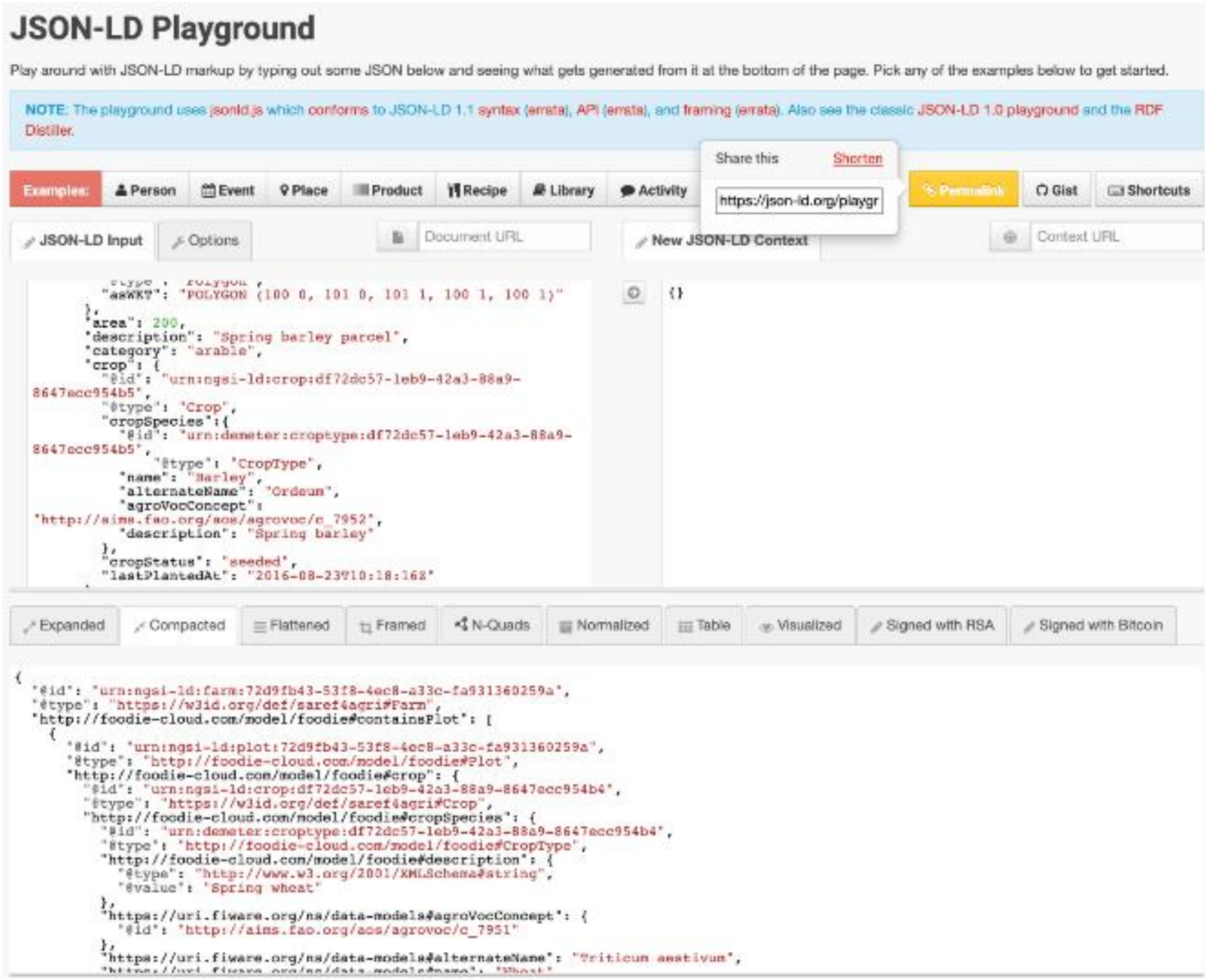

Figure 20. Validation of a simple farm example using the json-ld playground

The second, advanced level, of validation is to validate if content is semantically correct. This is done by evaluating the content against the AIM SHACL shapes graph[24]. In order to do this, there are different tools available:

- SHACL playground[25]. However, this tool is not checking some restrictions (e.g., datetime format). New

[24] https://raw.githubusercontent.com/rapw3k/DEMETER/master/models/SHACL/demeterAgriProfile-SHACL.ttl
[25] https://shacl.org/playground/

work was moved to a library[26].

- Apache Jena SHACL[27]
- Astrea Web Service[28]. This is a service under testing but provides a good basis for reusing.
- (recommended) PySHACL[29] command line tool

We recommend the PySHACL tool because, in addition to using the SHACL shapes graph, you can also include an ontology containing extra ontological information to mix into the data graph. In our case, this ontology is AIM itself. You can also specify different input format (e.g., JSON-LD, turtle) for both graphs. So, for example, if we validate example “pilot5.2-afc-observation-point-simplified.jsonld” using only the AIM SHACL shapes graph, we get an error that the value of “hasGeometry” property should be a “Geometry” (see table below). However, AIM defines that a “Point” is a subclass of “Geometry”, and so if we include AIM into the mix for the validation, the result is that it is valid (see the second table below).

```
./pyshacl --imports -s
https://raw.githubusercontent.com/rapw3k/DEMETER/master/models/SHACL/demeterAgriPro
file-SHACL.ttl -i rdfs -a -j -df json-ld -f human pilot5.2-afc-observation-point-
simplified.jsonld
Validation Report
Conforms: False
Results (1):
Constraint Violation in ClassConstraintComponent
(http://www.w3.org/ns/shacl#ClassConstraintComponent):
	Severity: sh:Violation
	Source Shape:
<https://astrea.linkeddata.es/shapes#3f6891594ac2d163a004bec00f8db48a>
	Focus Node: <http://www.w3id.org/afarcloud/poi?lat=45.75&amp;long=4.85>
	Value Node: <http://www.w3id.org/afarcloud/pCoord?lat=45.75&amp;long=4.85>
	Result Path: geo:hasGeometry
	Message: Value does not have class geo:Geometry
```

```
./pyshacl --imports -s
https://raw.githubusercontent.com/rapw3k/DEMETER/master/models/SHACL/demeterAgriPro
file-SHACL.ttl -e https://w3id.org/demeter/agri -i rdfs -a -j -df json-ld -f human
pilot5.2-afc-observation-point-simplified.jsonld
Validation Report
```

26 https://github.com/zazuko/rdf-validate-shacl
27 https://jena.apache.org/documentation/shacl/
28 https://astrea.linkeddata.es/swagger-ui.html
29 https://pypi.org/project/pyshacl/

```
Conforms: True
```

11.2 Types of Data modelled via AIM across the pilots

This section lists the types of data that are used across the pilots and are modelled based on AIM. This information has been originally collected from WP5 at the beginning of the project and has since then been updated twice in the process of stakeholder requirement extraction. Moreover, we included a respective question in the questionnaire circulated during the developer workshop conducted 29-31/3/2021, to collect the data types used across the DEMETER pilots. The available options in these questions have been the following:

- Farm data (e.g., field data, field status, soil data, Crops/treatment/fertilisation data, farm input data,energy consumption data, ...)
- Earth Observation Data (e.g., satellite data, remote sensing imagery, soil maps, vegetation indices,such as NDVI, EVI, NDRE, NDMI.)
- Meteorological data (e.g., temperature, humidity, wind speed/direction, solar radiation, pressure, etc.)
- Agricultural machinery data (e.g., engine data, fuel consumption, emissions, exhaust gas, NOx-conversion, exhaust temperatures, ...)
- Representation of data quality metrics
- Field Operations data (irrigation, fertilisation, soil tillage)
- Livestock data
- Traceability data (transport)
- Indicators, benchmarkings and KPIS
- Financial farm data
- Farmer information
- Other: (please specify)

Farm data and Meteorological data are the most popular data types used across most of the pilots that are modelled based on AIM or are translated to/from AIM. Moreover, additional types of data introduced by the pilot developers herewith are: Soil data, Apiary farm-related data and Milk robot data.

The following diagram presents a summary of the responses of pilot developers to the respective survey question.

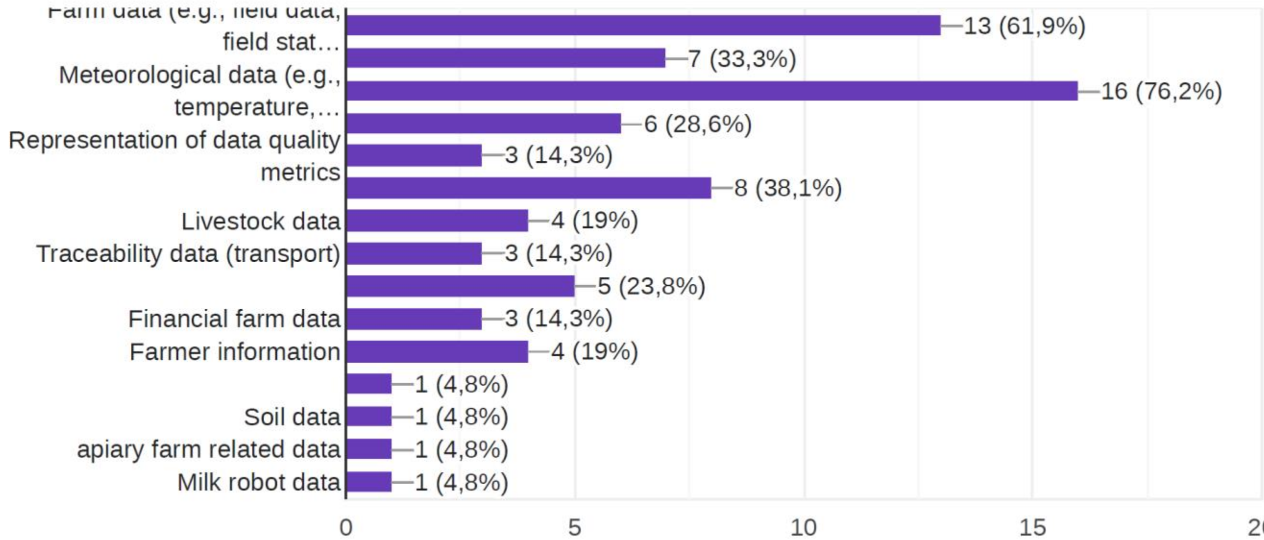

Figure 21. Summary of responses of pilot developers to the survey question: "*Do you manage in the pilots the fields geometry?*"

In the same questionnaire, another question has been included to capture the pilot developer responses on whether their pilot requires the management of fields' geometry. The feedback obtained indicates that the majority of the pilots are forced to deal with field geometry data regurarly or in specific occasions. A summary of the responses obtained is presented in the diagram below.

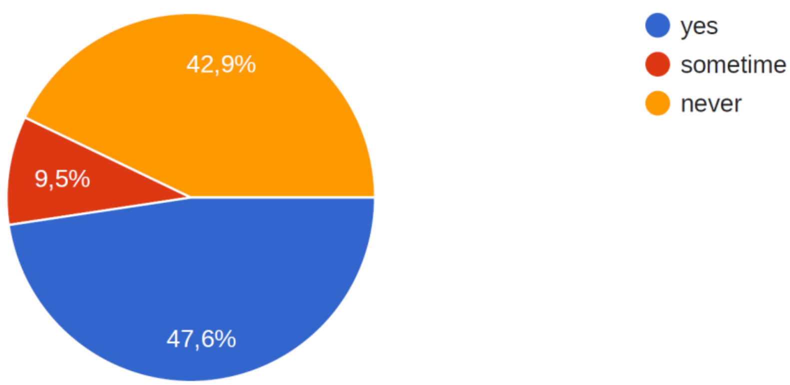

Figure 22. Summary of responses of pilot developers to the survey question: "*What type of data are you representing using AIM (or plan to do so)?*"

11.3 Pilot existing approaches for data modelling/semantics and respective AIM wrappers

This section aims to elaborate on the approaches for data modelling/semantics that pilots use inherently, before translating these to AIM and the respective AIM wrappers that are in place. This information has been regularly collected from WP5 since project start. Moreover, we included a respective question in the questionnaire circulated during the developer workshop conducted 29-31/3/2021, to collect feedback on the data model/vocabulary or standard or data format that the DEMETER pilots use that needs to be translated to AIM and vice-versa. The obtained responses are listed below:

- Semantic Sensor Network (SSN)
- ISOBUS
- ROS
- NGSI-LD
- FADN
- SensorThings
- ICAR NCDX - Global Standard for Livestock data - Milking robots
- Json
- PostgreSQL
- CSV, xml
- AFarCloud Data Model
- Georeferenced images (format such as Geotiff)
- Ex-Machina platform data model
- Custom formats on field information and Earth Observation timeseries
- Custom apiaries related vocabulary and model

The implemented AIM wrappers and developed software that facilitates the translation of pilot data formats to AIM and vice versa are available on the DEMETER gitlab under: https://gitlab.com/demeterproject/wp2/semanticinteroperability

11.4 Further AIM extensions requested by pilots for next release [ICCS]

AIM needs to be further extended in order to completely cover the needs of pilots for proper integration of their data and a number of modules is planned to be implemented for the next release. The following are some modules that have already been identified:

- **Agricultural Measurements:** This module shall model the whole system of taking measurements via imagery, including parameters of the image like width, height, source, format and resolution and also measured quantities like CO2 or chlorofyll levels
- **Irrigation Processes:** This module is planned to represent a plot irrigation system, including drips,

sprinklers, etc. Some needed properties are sytem's reliability, emitters, distance, fluid flow rate, advection and water conductivity.

- **Decision-making & Forecasting:** This module is needed as a generic ontology for representing decision support systems forecasting output. Some of the predicted values or decided actions are irrigation needs, crop water needs, evapotranspiration, rainwater forecast, average soil moisture, plant status and plant anomalies.
- **Various extensions:** It is yet to be decided whether the following terms need to be inserted in new ontologies or it is enough to extend existing modules. These terms are percolation, plant diameter, crop coefficient, yield, DEF level and dosing temperature, fuel temperature, engine oil pressure and temperature, engine coolant level and pressure, SCR in/out temperature, NOx in/out

Finally, there have been a few additional comments indicated by the pilot developers in response to the same survey. These indicate further changes over the AIM release 2 and are listed below:

- All users that use e.g., EO data in AIM format, could work more closely together to jointly discuss AIM solutions
- There are other properties and objects that are not define in AIM yet, that will be requested for and introduced in the final AIM release.
- We will use AIM only where we need to exchange data with other Demeter Components.
- In Pilot Round 2, we will need to integrate additional data (coming from the sensors Farm A) so our model will have to be integrated with new information, but also in this case custom data.
- Due to the significant size increase of image data when converted to base64 encoding in order to include them to AIM json format, it might be advisable to work out a method that would allow to include links where such data can be downloaded/obtained from rather than to include them in AIM format.
- We need to federate related multiple authentication OAuth2 services

12 Conclusions

This deliverable describes in detail the second release of the DEMETER Common Data Models and Semantic Interoperability Mechanisms and presents the DEMETER Agricultural Information Model (AIM) (Release 2), as updated after the experience gained from the implementation of DEMETER and AIM during the first round of DEMETER pilots. It initially presents the methodology used in order to drive the revisions together with an overview of the changes and updates to the AIM model. Then it presents the questionnaire findings regarding AIM changes, issues and missing concepts submitted by the DEMETER partners and the way they contributed to the revisions of AIM. Subsequently, it presents the revised technical requirements initially extracted by Task 2.1 in D2.1 and, here, revised depending on the degree of coverage by the implementation of AIM and the various enablers; these technical requirements state what information AIM needs to be able to represent, as well as requirements regarding the interoperability with existing systems, ontologies and data formats. Following this, this deliverable presents in detail the updates made to the second release of the DEMETER Agricultural Information Model (AIM) compared to what was presented in Deliverable 2.1. While AIM still adopts a modular approach with several components/layers, i.e., the AIM core metamodel, the AIM cross-domain ontology, the AIM domain-specific ontologies and the AIM metadata schema, all of which were initially presented in Deliverable 2.1, these have been revisited to a significant degree and a new layer (the domain specific ontologies) has been added to represent data that are required by specific pilots and are not covered by the domain specific ontologies. Next, it presents an updated semantic mapping of, and the interoperability support between, the DEMETER AIM and several existing ontologies and dominant agri-food systems. Following, it presents the implementation of the DEMETER AIM together with elements such as the mappings implemented and the tools used during the implementation process as well as the usage of AIM across the pilots, examining aspects such as: AIM usage guidelines for pilot developers, pilot existing approaches for data modelling/semantics and the respective AIM wrappers, the datatypes modelled and further AIM extensions and pilot-specific ontologies that are requested by pilots.

Regarding the work still to be performed and the expansion of AIM with new concepts, we present here a quick summary of the plan for future work. In short, several AIM pilots require information pertaining to the traceability of food products, therefore, the traceability information described in the FoodOn ontology will be useful for this process and is going to be incorporated into the DEMETER AIM. If needed, we also plan to only incorporate any information from eCrop that might not be encompassed already by the FoodOn additions. GS1 is already used by some projects (e.g., eCrop) to tag information. To this end, it would be useful to (at least partially) incorporate the standard into our traceability extension to AIM before the end of the project. Finally, we are examining whether it makes sense to incorporate concepts from AFarCloud, especially as pertaining to sensors, actuators and devices, vehicle and mission planning related data, or extend the animal and crop data models. For more details, please refer to Annex A, where an addendum to the state of the art is presented examining the projects aforementioned and also describing which concepts from these projects might be of interest for the future extension of AIM.

The content of this deliverable is the result of collaborative work of partners in every single work package of the project, as their experiences and requests for AIM changes drove the revisions presented in this deliverable. In particular, while most of this document has been prepared by Task 2.1 (which is responsible for this deliverable), Sections 6 and 7 have been prepared based on the input to questionnaires and to the technical requirements by several partners, while the usage of AIM in the pilots presented in Section 11 have been contributed by several pilot developers.

It should also be highlighted that the core outcomes presented in this deliverable have been used as the main input to put together an extended book chapter [RP2021] this has been accepted and will be published before the end of this year.

This deliverable contributes to the achievement of Milestone 6 (DEMETER Enablers, Hub, Spaces and Applications Release 2) planned for June 2021. The AIM revisions presented in this deliverable will influence a number of deliverables that follow, i.e.:

- D2.4 DEMETER data and knowledge extraction tools – Release 2 (May 2021)
- D4.3 Decision Support, Benchmarking and Performance Indicator Monitoring Tools – Release 2 (May 2021)
- D4.4 Decision Enablers, Advisory Support Tools and DEMETER Stakeholder Open Collaboration Space - Release 2 (June 2021)

Finally, following this, we conclude by examining a number of additional ontologies and systems that could be made interoperable with the DEMETER AIM. This constitutes ongoing and future work that is scheduled for investigation towards the final AIM release that will be presented in Deliverable D2.5, to be released in October 2022. This work intends to further develop AIM and extend its interoperability support to address additional data types and to address new needs of the DEMETER pilots about additional concepts not covered by the second AIM release. To this end, Annex A elaborates on a number of state of the art models / ontologies that can be used to extend AIM, while preliminary approaches about potential integration of these with AIM are also discussed.

13 References

[AT09] Athanasiadis, Ioannis N., et al. "Ontology for seamless integration of agricultural data and models." Research conference on metadata and semantic research. Springer, Berlin, Heidelberg, 2009.

[BR20] Brown, David, et al. "Data synthesis for crop variety evaluation. A review." Agronomy for sustainable development 40.4 (2020): 1-20.

[DO18] Dooley D.M. et al. "FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration" (2018), Nature.

[eCrop18] United Nations Centre For Trade Facilitation and Electronic Business (UN/CEFACT), "BUSINESS REQUIREMENTS SPECIFICATION (BRS) for eCROP" v1.0, approved January 2018. https://www.unece.org/fileadmin/DAM/cefact/brs/BRS_eCROP_v1.pdf

[ETSI19] ETSI GS CIM 009 V1.1.1 (2019-01), "Context Information Management (CIM); NGSI-LD API", https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_cim009v010101p.pdf

[GR13] Grimnes, G. A., Kiesel, M., & Bernardi, A. (2013). Ontology-Based Mobile Communication in Agriculture. KI-Künstliche Intelligenz, 27(4), 335-339.

[GS1] GS1 Web URI Structure Standard, GSI Digital Link Standard documentation https://www.gs1.org/standards/Digital-Link/1-0

[IG14] Iglesias, N., Bulacio, P., & Tapia, E. (2014). Enabling powerful GUIs in ISOBUS networks by transparent data compression. Computer Standards & Interfaces, 36(5), 801-807.

[ISO11783-11] ISO 11783-11, Mobile Data Element Dictionary, Online Data Base Info. https://www.isobus.net/isobus/images/ISO-11783-11-Online-Data-Base-Info.pdf

[KR15] Kruize, J.Willem, et al. "A Farm Information Model for Development and Configuration of Interoperable ICT Components to support Collaborative Business Processes" Food System Dynamics (2015).

[MA13] Matteis, Luca, et al. "Crop ontology: vocabulary for crop-related concepts." (2013).

[OGC16] "OGC OpenSearch Extension for Earth Observation", Pedro Gonçalves, Uwe Voges (2016), http://docs.opengeospatial.org/is/13-026r8/13-026r8.html

[OWL12] OWL 2 Web Ontology Language, New Features and Rationale (Second Edition), W3C Recommendation 11 December 2012, https://www.w3.org/TR/owl2-new-features/#F12:_Punning

[PR21] Palma R., Roussaki I., Döhmen T., Atkinson R., Brahma S., Lange C., Routis G., Plociennik M., Mueller S. (2021). "Agriculture Information Model" in D. D. Bochtis, C. Sørensen, S. Fountas, V. Moysiadis and P. M. Pardalo (Eds). Information and Communication Technologies for Agriculture—Theme III: Decision. Springer (Book Chapter accepted for publication)

[SSN17] Semantic Sensor Network Ontology, W3C Recommendation 19 October 2017 (Link errors corrected 08 December 2017), Armin Haller, Krzysztof Janowicz, Simon Cox, Danh Le Phuoc, Kerry Taylor, Maxime Lefrançois, Rob Atkinson, Raúl García-Castro, Joshua Lieberman, Claus Stadler, https://www.w3.org/TR/vocab-ssn/

[TI20] Time Ontology in OWL, W3C Candidate Recommendation 26 March 2020, Editors: Simon Cox, Chris Little, Jerry R. Hobbs, Feng Pan, https://www.w3.org/TR/owl-time/

[W3C] W3C Working Group Note 06 December 2016 (Link errors corrected 7 Mar 2017), Anne van Kesteren, Sam Ruby, https://www.w3.org/TR/url/

Annex A: State of the Art Review (addendum)

In this annex, we present an examination of several ontologies that are planned to be included (or rather have their concepts inserted) next in the AIM model together with a plan of what concepts are to be incorporated into the DEMETER AIM. Therefore, this annex gives more information on the future plan for AIM revisions and potential additions to be considered for implementation in the final AIM release, as summarized in the future work discussion in Section 12.

A.1 FOODON Ontology

The FOODON project [DO18][30] aims to build a comprehensive and easily accessible global farm-to-fork ontology about food, that accurately and consistently describes foods commonly known in cultures from around the world. FOODON addresses food product terminology gaps and supports food traceability (following the GS1 standard[31]). The FOODON ontology covers basic raw food source ingredients, process terms for packaging, cooking and preservation, and an upper-level variety of product type schemes under which food products can be categorized. It is built to interoperate with the OBO Library and to represent entities which bear a "food role". Initially, FoodOn was based largely on LanguaL[32], a descriptive food indexing system that has 14 facets for describing food source plant and animal organisms, food preservation, cooking, packaging, consumer groups, labelling, etc. and has since been extended with new concepts derived from a number of other existing ontologies.

To this end, it has concepts that relate to all types of food types as well as their component, the processing they have undergone (e.g., including a list of chemical agents that could have been used in the food processing or recording whether a few has undergone heating and other types of processing) and also has a list of countries (and in some cases provinces or areas) from which specific food products originate. Overall, the full FOODON ontology has an extensive vocabulary describing anything related to all these.

As can be seen in Figure 23 below, we present a food product diagram based on the FoodOn ontology; this depicts several of the classes and relationships defined in it and contains information such as the source of food used to create the product, what kind of processing has been done in order to produce it, but also other information such as the packaging it is in, as well as cultural origins as well as consumer groups for it.

[30] Also see the FOODON website: https://foodon.org/
[31] This standard is discussed and can be found later in this report.
[32] http://langual.org/

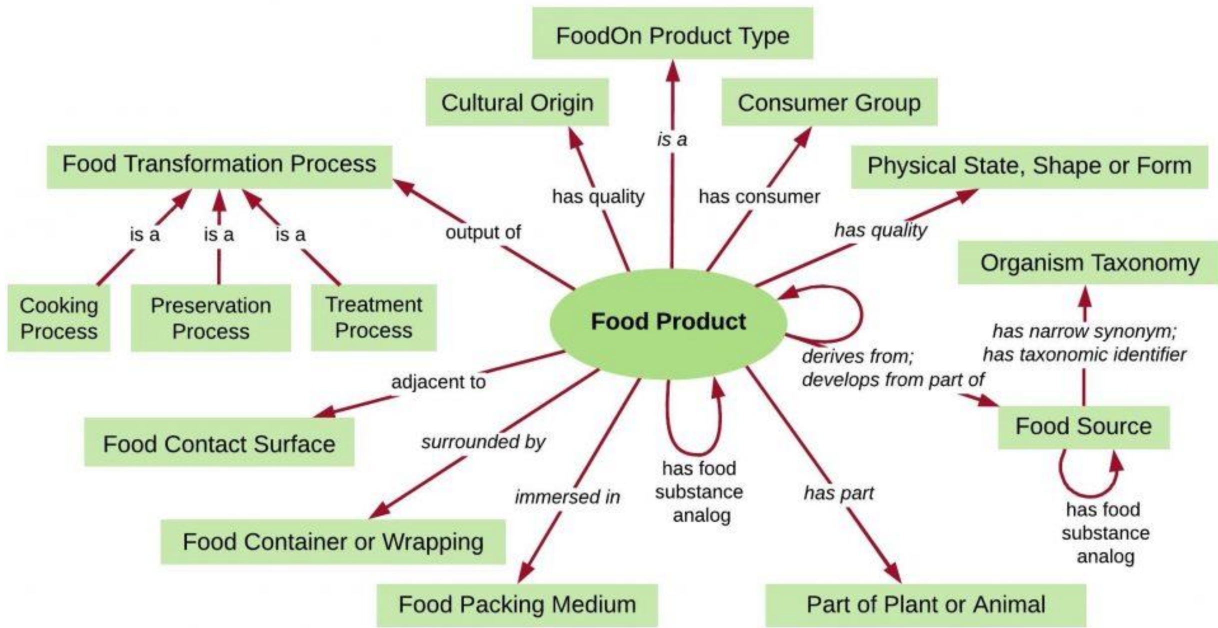

Figure 23. Food product diagram based on the FoodOn ontology depicting several of the classes and relationships defined in it.

More specifically, the ontology covers terms for the origin of food sources, the processing and cooking that a product has undergone (e.g., chemical or heat processing, drying etc. to mention a few), even covering how it is packaged (the container or the wrapping of this product). Of course, it also includes information regarding where the material (e.g., meat or plant etc.) for the food originated, any additives needed for the processing as well as detailed information regarding the processing (e.g., in chemical treatment what types of chemical and additives are used, or to which degree it was heated during a heating procedure). As such it is useful to any application where it is important to keep track of where a food product originates from (including the source of it; e.g., milk from cows in a specific area), the chemical used for processing and processes it has undergone to get the final product. In case where the product uses multiple simpler

In Figure 24 below, we present an example usage of the FoodOn ontology when used to describe food product information for corn flakes showing the material from which they are made (i.e., corn which has then been processed to corn meal), that it's a type of cereal and that it has undergone milling and flaking processes and finally has been produced as the output of a dehydration procedure.

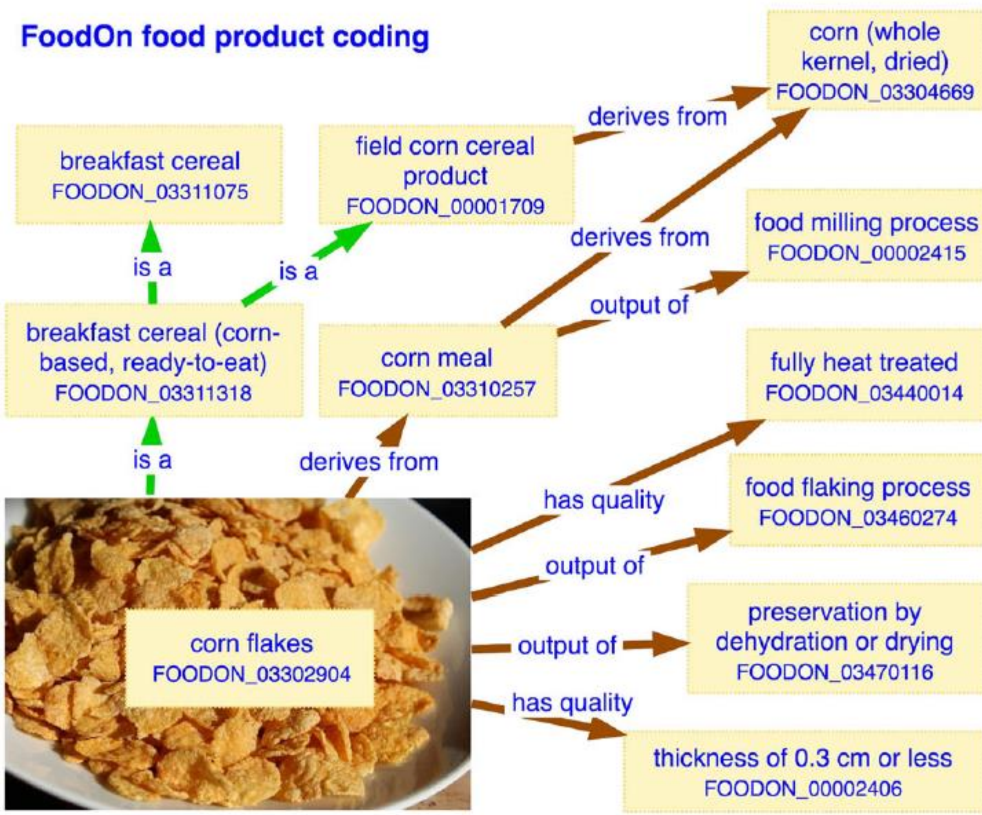

Figure 24. An example usage of the FoodOn ontology describing food product coding for corn flakes

This ontology generally categorizes the origins of each food source by defining named individuals for each country (and furthermore US state) in the world; however, for what we need in the DEMETER pilots, it might be more appropriate to use GPS or factory and farm information in order to further specify where each food source originates from or facilities where food processing took place. But goes even further when the food product is a complex food, detailing the basic plant or animal organism food items used to produce this more complicated food. These basic food items could be a whole organism (e.g., sardine), a product derived from an organism (e.g., meat from a cow, in which case the processing to derive the meat would also be described) or it could be part of an organism (e.g., the harvested part of the mushroom).[33]

The ontology can also be used for and provides the vocabulary for nutritional analysis, including chemical food components which are factors in diet, health and plant and animal agricultural rearing research. For example, Figure 25 describes the general structure used by the currently evolving FOODON ontology to describe the nutritional analysis of a specific food product. We can observe that it includes concepts such as the origin, the harvested material, chemicals, processing and other related information regarding a specific food material.

[33] These terms come from other existing ontologies and are being reused by FOODON.

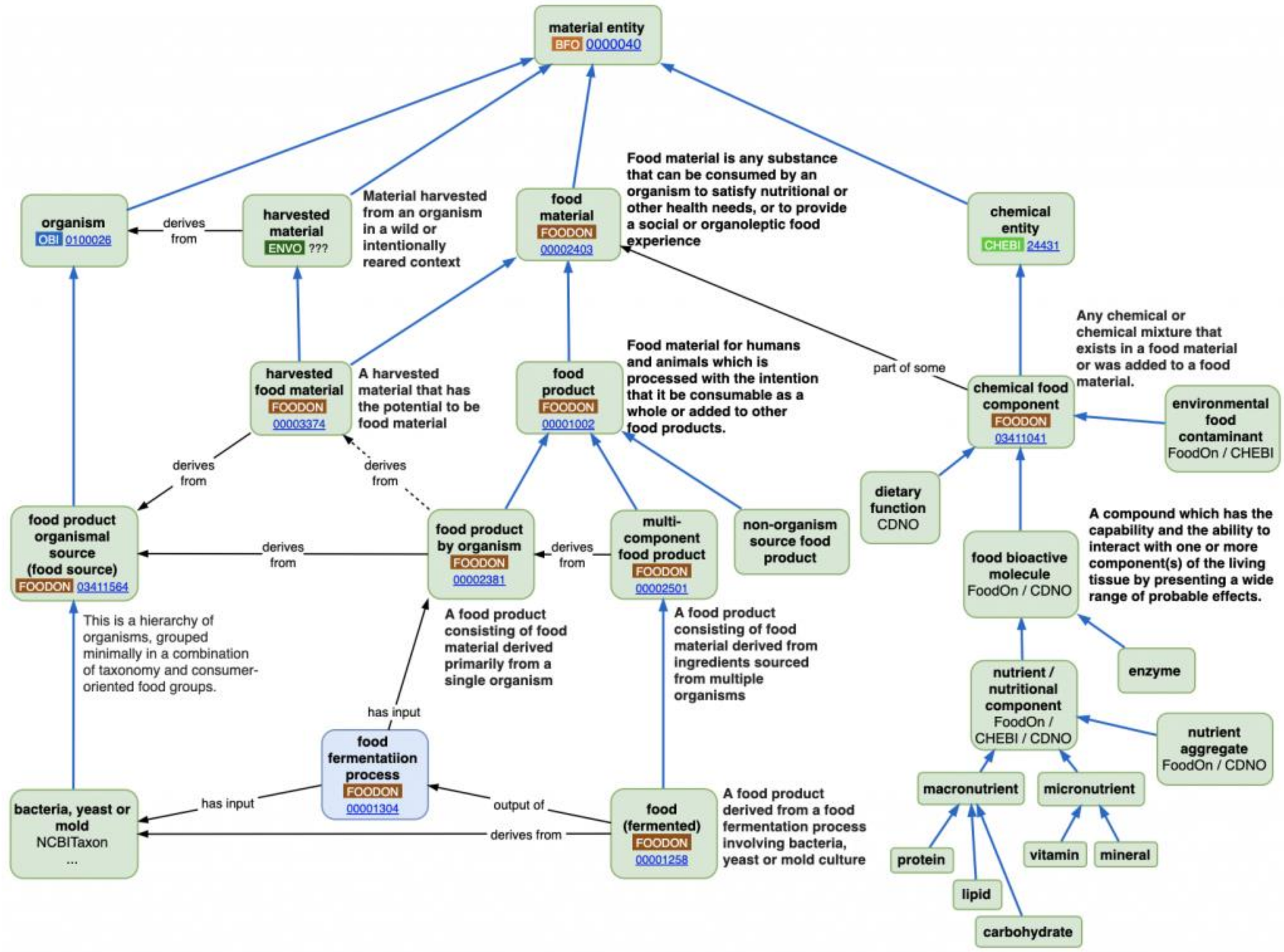

Figure 25. The general structure of the FOODON ontology to describe a food's nutritional analysis

The ontology contains very detailed information of the origin (e.g., from an animal or harvesting a specific crop etc.) and of any food production. In Figure 26, the origin of milk is presented as an example: not only is the source (that milk comes from the mammary gland of a cow) but also the main processing that milk usually undergoes (e.g., homogenized and pasteurised) and also undergoing a drying process in this specific case.

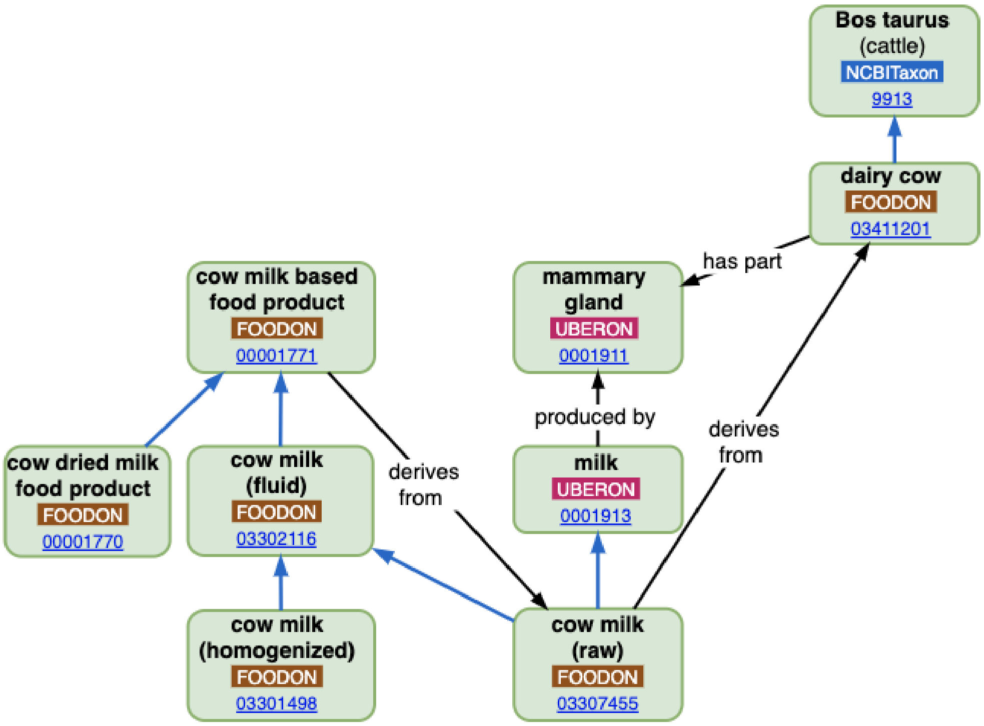

Figure 26. An example usage of the FOODON ontology to describe the milk origin

<u>Incorporation of FOODON into the DEMETER AIM:</u>

Several AIM pilots require information pertaining to the traceability of food products, e.g., from the animal that produces the milk and the conditions under which this animal was raised; for example its location, the farm to which it resides, the food that it is fed, the stress of the animal and a number of other physiological features described in D5.3 regarding the data and processing that the DEMETER pilots require to do in particular those of cluster 5. Additionally, the processing and chemicals or procedure (e.g., pasteurization) that the product undergoes, the packaging and tracking information would also be useful.

To this end, the traceability information described in the FoodOn ontology will be useful for this process and is being incorporated into the DEMETER AIM. Although, it will limit the amount of information (at least for the time being) to what is required by the pilots and will not include extensive information on all chemicals and processes that a product may undergo. Other information that is too detailed and will not be incorporated is detailed anatomy information, e.g., that the cow has a mammary gland that produced the milk, as such detailed information would not be useful for most applications. In general, the model is extended to allow for the inclusion of the data that would be useful in order to keep a product passport information.

A.2 eCrop Initiative

The eCrop initiative is promoted by the United Nations Centre for Trade Facilitation and E-Business (UN/CEFACT) and its goal to facilitate the "exchange of production and cultivation data … for plant products". To this end, it has produced a number of XML schemas to exchange crop related data and especially as relating to business processes. Our main source of reference for eCrop is the Business Requirements Specification document of eCrop. [eCrop18]

Now, the data exchanged focuses mainly on the supply chain from the growers and producer organizations, to food processors all the way to the retail. The goal is to promote sustainable production farming, food quality and food safety. In more detail, the information exchanged includes:[34]

- Information about the crop fields, including production location, crop rotation schema, treatment, plant and animal health, soil and water situation, crop observation data, supplies (such as seeds or fertilizer), operation instructions and the operations logs.
- Information about the production of the crop (or product): e.g., crop protection, health agents, fertilizers applied, animal feed, use of energy and water.
- Geo-information and location data about the exact location of field operations.
- Information about labour.
- Information about certificates (such as GlobalGap or SPS certificates, farm or product).

This information is then to be used in the tracing and tracking of products with the aim to certify compliance with regulations and certifications, with government laws and best practices; also to track the supply chain of a product from production to processing to retail in order to provide traceability information as well as to provide this data to advisory services that would support the farmer in making the right decisions in nurturing his crop and in precision farming, in the correct mode of transport and then regarding the processing and packing of said product.

To this end, the following figure presents the scope and the actors of the eCrop initiative. The farmer (or FMIS) is in the center and information is exchanged between the farmer and contractors, buyers, government and certification authorities as well as advisors and suppliers in order to support the aforementioned use cases (such as supply chain tracking and tracing, compliance and cultivation support).

[34] Information taken from the eCrop BRS. [eCrop18]

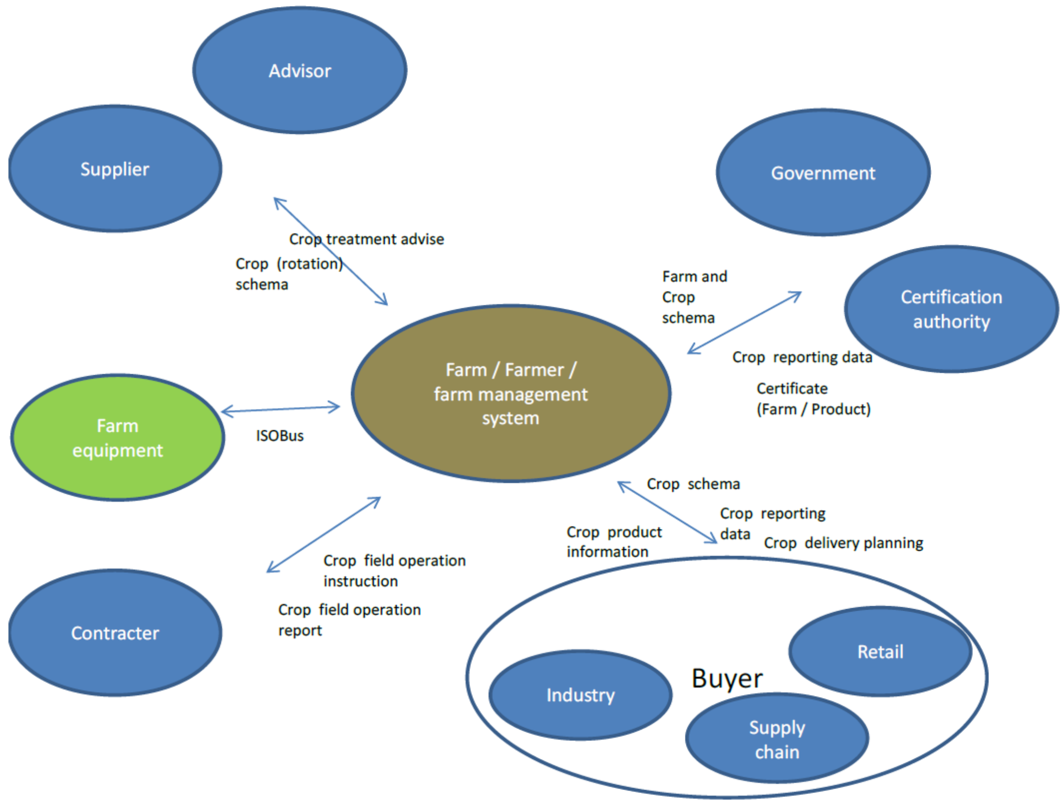

Figure 27. The scope of the eCrop initiative

Looking at the data models used by eCrop, the eCrop XML schemas have been influenced by the Crop Reference Data and related models developed by the Wageningen University and Research regarding related topics.[35] More specifically, it follows the rmAgro/drmAgro/drmCrop standardisation of the electronic data exchanged and relevant architecture.[36] In the following figure, the eCrop schema generic structure is presented. One can observe that it covers data pertaining to the crop grown in which plot/location, by which party, then the product that it becomes later in the supply chain, including information about the processing (and the organization that did the processing).

[35] For more information visit the Wageningen site: https://www.wur.nl/
Some references regarding the related work are the following: [AT09], [MA13], [KR15]; and a more recent review paper regarding the data synthesis of agri-data is [BR20].
[36] For details about rmAgro see section 5.7 of our previous deliverable D2.1.

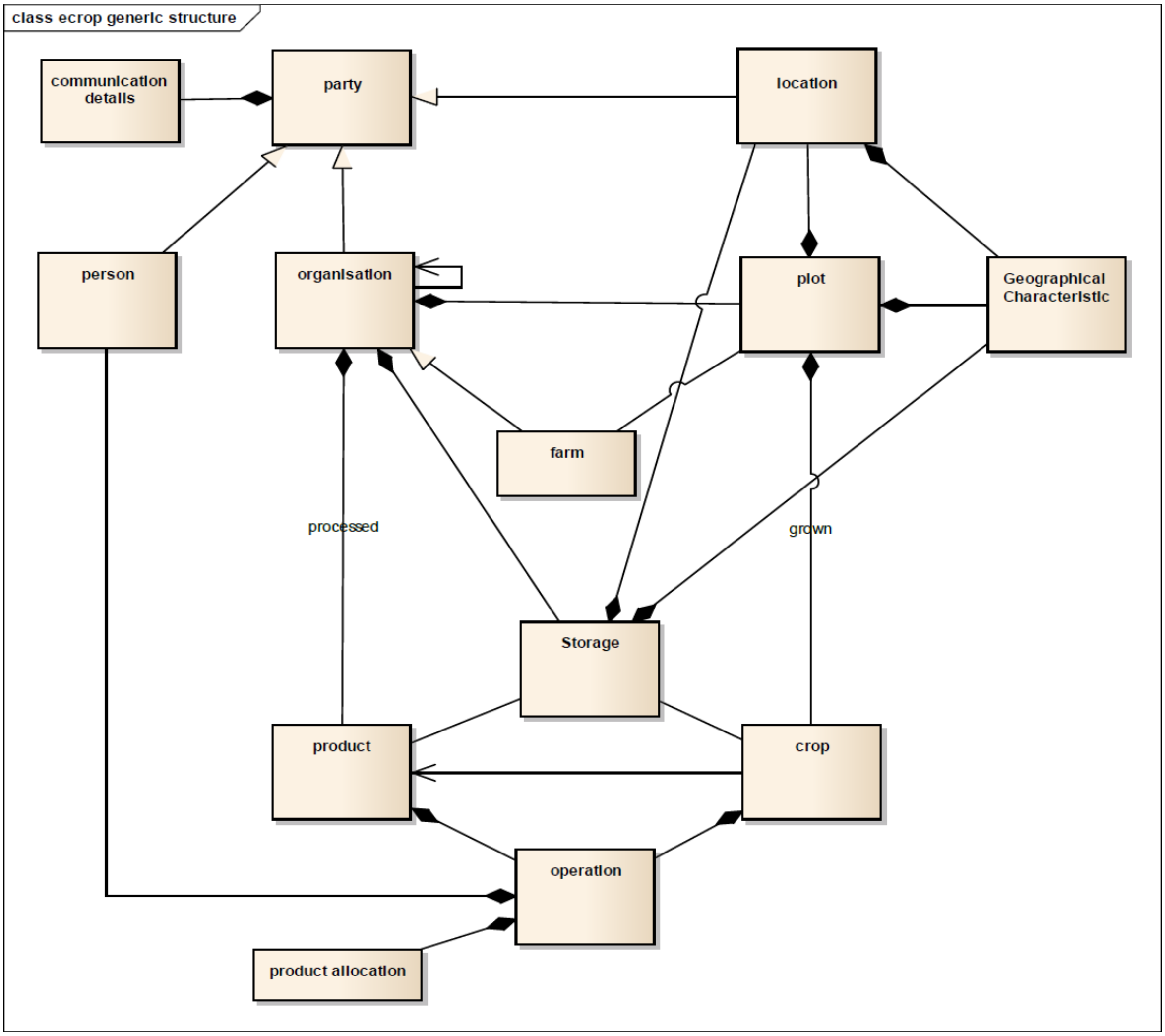

Figure 28. eCrop schema generic structure

In general, the eCrop XML schema provides information about crops, the people and organizations who grow them, process them, the actors involved and the processing involved in the business of the supply chain starting from the growing and treatment of a crop (or other food, plant etc.) in the field, to its processing (including information about the chemicals of said product) down to the final retail. As mentioned earlier, all this information is useful in order to ensure the sustainability and food safety which is a key objective of eCrop.

How this information may be used in an actual system that takes the eCrop data models and implements a supply chain tracing and tracking application (or ecosystem in case many of such applications interact) is displayed in the following figure. In it, an example usage case envisaged by eCrop is presented; the information regarding the growing processing packaging etc of a product from the producer/grower of the crops to the final processed retail product is tracked and logged using the appropriate repositories and then services can query and get the information they need to track using the appropriate discovery applications.

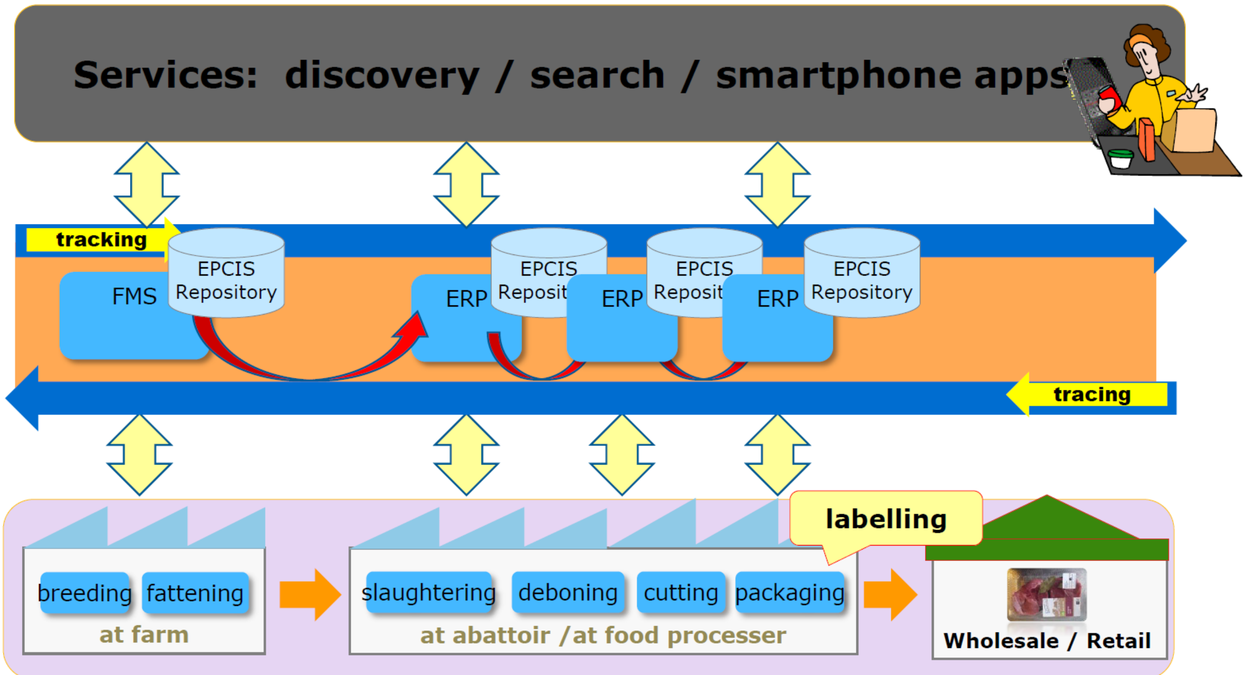

Figure 29. An example usage case envisaged by eCrop (source: GS1 Germany GmbH)

<u>Incorporation of eCrop into the DEMETER AIM:</u>

When presenting the integration of the FoodOn ontology concepts into AIM, we discussed that concepts of this would be imported into the DEMETER AIM to enable traceability and the creation of a food/product passport. The eCrop model supports a lot of similar business processes and the related information. However, at this stage we plan to only incorporate any information that might not be encompassed already by the FoodOn additions, and at this point this seems to be rather minimal information compare to what would already be added from FoodOn.

A.3 ISOBUS

ISOBUS is based on the ISO 11783 standard and focuses on machinery and implements for the agricultural and forestry industry including tractors. Besides the definition of communication protocols (including CANBus), it also defines a vocabulary to ease the communication and integration between machines coming from different manufacturers. Thus, ISOBUS aims to provide an open system for interoperability and standardize data exchanging between systems including sensors, actuators, control elements, information storage, and display units mounted on tractors or implements [IG14].

The ISOBUS data dictionary[37] (according to ISO11783-11) currently contains more than 640 entities related to 27 different device classes such as tractor, primary soil tillage, secondary soil tillage, planter/seeder, fertilizer, sprayer, harvester, root harvester, forage harvester, irrigation, transport/trailer, farmyard work, powered auxiliary unit, special crop, municipal work, sensor system, timber harvester, forwarder, timber loader, timber processing machine, mulcher, utility vehicle, slurry applicator, and feeder/mixer.

The entities are described with a definition, corresponding typical device class(es), unit, and further communication details such as the CANBus range. Thereby, the Data Dictionary Entities reflect the values of sensors that are part of those devices/electronic elements. Examples for such entities are lifetime average fuel consumption per area, total application of nitrogen, actual volume per area application rate as [mm^3/m^2], loaded total count, setpoint tire pressure, chopper engagement total time, actual engine speed, etc.The current list of entities is available as a pdf[38] or text[39] file and the entities can additionally be accessed in Json [ISO11783-11] as seen in Figure 30. Moreover, the ISOBUS group defines a process to add new entities to this standard list [GR13].

DDI URL and Jason Link

The online data base supports Jason File's and a secure link to DDI numbers.
The following information is provided to access the DDI information directly from the Internet:

DDI Link
https://www.isobus.net/isobus/dDEntity/detail/*DDI-Number*
Example with DDI 70
https://www.isobus.net/isobus/dDEntity/detail/70

Jason File Support
https://isobus.net/isobus/dDEntity/detailJson/*DDI-Number*
Example with DDI 156
https://isobus.net/isobus/dDEntity/detailJson/156

Figure 30. Json file support [ISO11783-11]

37 https://www.isobus.net/isobus/
38 https://www.isobus.net/isobus/site/exports?view=export
39 https://www.isobus.net/isobus/exports/completeTXT

For DEMETER, including ISOBUS could be beneficial as the standard also aims to increase interoperability and involves already input coming from several companies (more than 1000 manufacturer IDs are listed). Nevertheless, integration to AIM could be challenging because ISOBUS does not provide a whole ontology, the only mapping of entities is done regarding the device classes.

A.4 GS1 Digital Link Standard

GS1 is standardisation body defines a wide range of identifiers that underpin the supply chain and retail industry. The GS1 Digital Link standard [GS1] references a number of third-party standards from the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).

This work has been motivated by a need in the supply chain to move to 2D barcodes that can carry more information than just the GTIN[40]; the problems of multiple barcodes causing scanning errors through conflicts which suggests a need for a single but multipurpose barcode; the growing expectation among consumers that more information is available online about the products they're considering buying; the brand owner concept of the pack as a media channel linking to multimedia experiences, and more.

The main result of this standard, it the possibility to use GS1 identification keys consistently within Web addresses as well as within barcodes containing Web addresses, such that a single identification approach can support both product identification for supply chain applications and a link to online material for consumer and business partner interactions.

At its most basic, the structure of a GS1 Digital Link URI is composed of a three key parts: resolver, primary key name and primary key:

https://{resolver}/{primaryKeyName}/{primaryKeyValue}

In addition, there are also parameters allowed, as indicated in Table 11. This can be extended even further since a GS1 Web URI can be constructed in any domain name, may contain additional key/value pairs in the query string and so on. This flexibility is a deliberate feature of the standard to support its use in as many scenarios as possible. However, in some contexts it is necessary to identify a single preferred version of the URI. This is defined in [RFC 6596] as the canonical URI. Since the GS1 Web URI encodes element strings as defined in the GS1 General Specifications [GENSPECS] such as that

https://example.com/gtin/614141123452

and

(01)614141123452

are equivalent, and short string alternatives for a subset of GS1 keys, define the canonical URI as follows:

[40] Global Trade Item Number

- the domain name SHALL be id.gs1.org;
- convenience string equivalents for AIs SHALL NOT be used;
- non-GS1 key=value pairs SHALL NOT appear in the query string.

Table 11. GS1 Digital Link URI structure details

URL value	Comment	Data type
Resolver	the domain name	String
Primary key name	This is application identifier and it indicates use of the URL, with predefined values: “ship-to”, “gtin”, etc (please look Section 5.1.2 of the GS1 [GS1] for more details).	String
Primary key value	The value pair for that corresponds to the key name, e.g., ship-to ID, or gtin ID	String
Parameters	Parameters that can be added on the end of the url	Array string

Incorporation of GS1 standard into the DEMETER AIM:

GS1 is already used by some projects (e.g., eCrop) to tag information. To this end, it would be useful to (at least partially) incorporate the standard into our traceability extension to AIM and there is a plan to be incorporate before the end of the project. GS1 could be useful to extend all tracking information used for traceability and supply chains, as this is what it is designed for, i.e., to represent on physical products (with a barcode potentially) the product origin as well as information about the material (e.g., source food) that it has been produced from.

A.5 The AFarCloud Ontology

The AFarCloud project[41] aims to provide a distributed platform for autonomous farming that allows the integration and cooperation of agriculture Cyber Physical Systems in real-time in order to increase efficiency, productivity, animal health, food quality and reduce farm labour costs. This platform is integrated with farm management software and supports monitoring and decision-making solutions based on Big Data and real time data mining techniques. Components of the AFarCloud project will be used in the pilot 5.2 of Demeter. The AFarCloud ontology[42] is divided into several domains that match the different kinds of data that are processed in the project: the Farm domain, the Robotic Vehicle domain, the Sensor domain and the Mission domain. These domains are described in detail below.

[41] http://www.afarcloud.eu/
[42] www.w3id.org/afarcloud

The Farm domain is related to entities and animals present in a farm and their main characteristics. Examples include livestock categories and general information, e.g., if they are used for dairy or beef production, their feed, transactions and health condition. The farm domain also models the crops that are grown in the farm. The crops model is based on the information model used in the FOODIE project[43]. The main concepts and the relationships of the Farm domain are described in Figure 31.

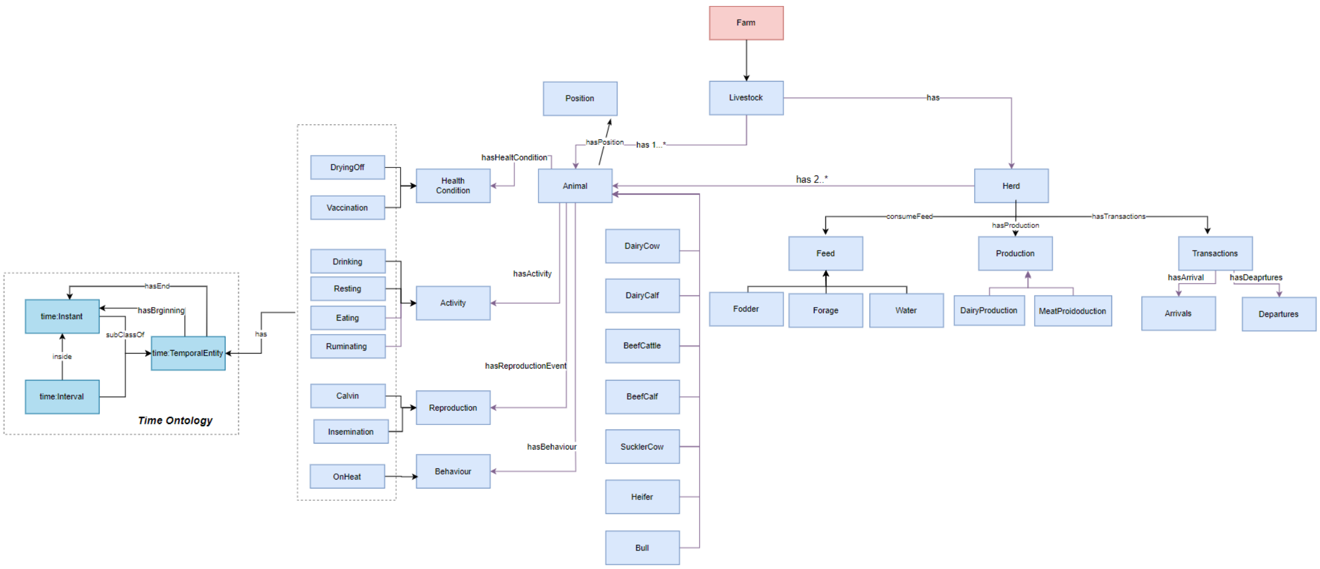

Figure 31. Structure of the Farm domain of AFarCloud

The Robotic Vehicle domain models the vehicles that could be used in AFarCloud missions and defines a taxonomy for all of them. In AFarCloud, vehicles can travel by air (e.g., UAV) or operate on the ground (e.g., Tractor, UGV). This domain is based on the Robotic Service model of the H2020 project SWARMs[44]. In this case, the SWARMs model has been extended for ground and aerial vehicles. Figure 32 describes the relations of the Robotic Vehicle domain and Figure 33 represents the hierarchy of robotic vehicles.

[43] http://www.foodie-project.eu
[44] http://www.swarms.eu/

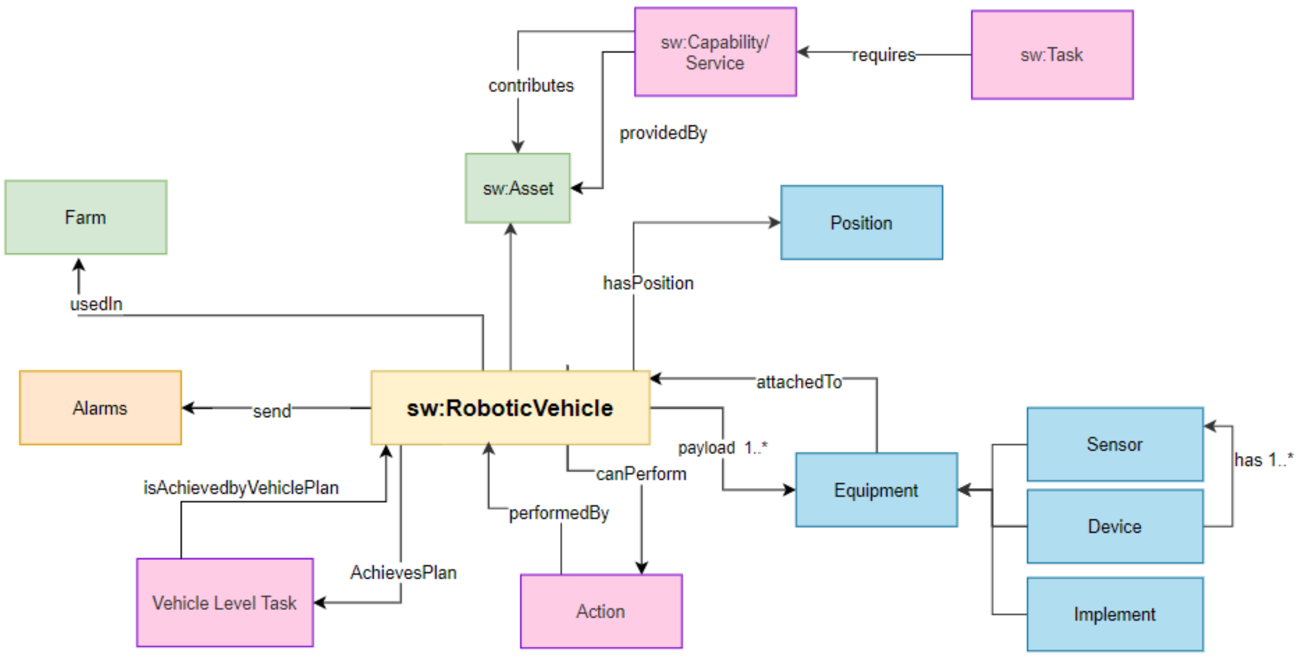

Figure 32. Structure of the Robotic Vehicle domain of AFarCloud

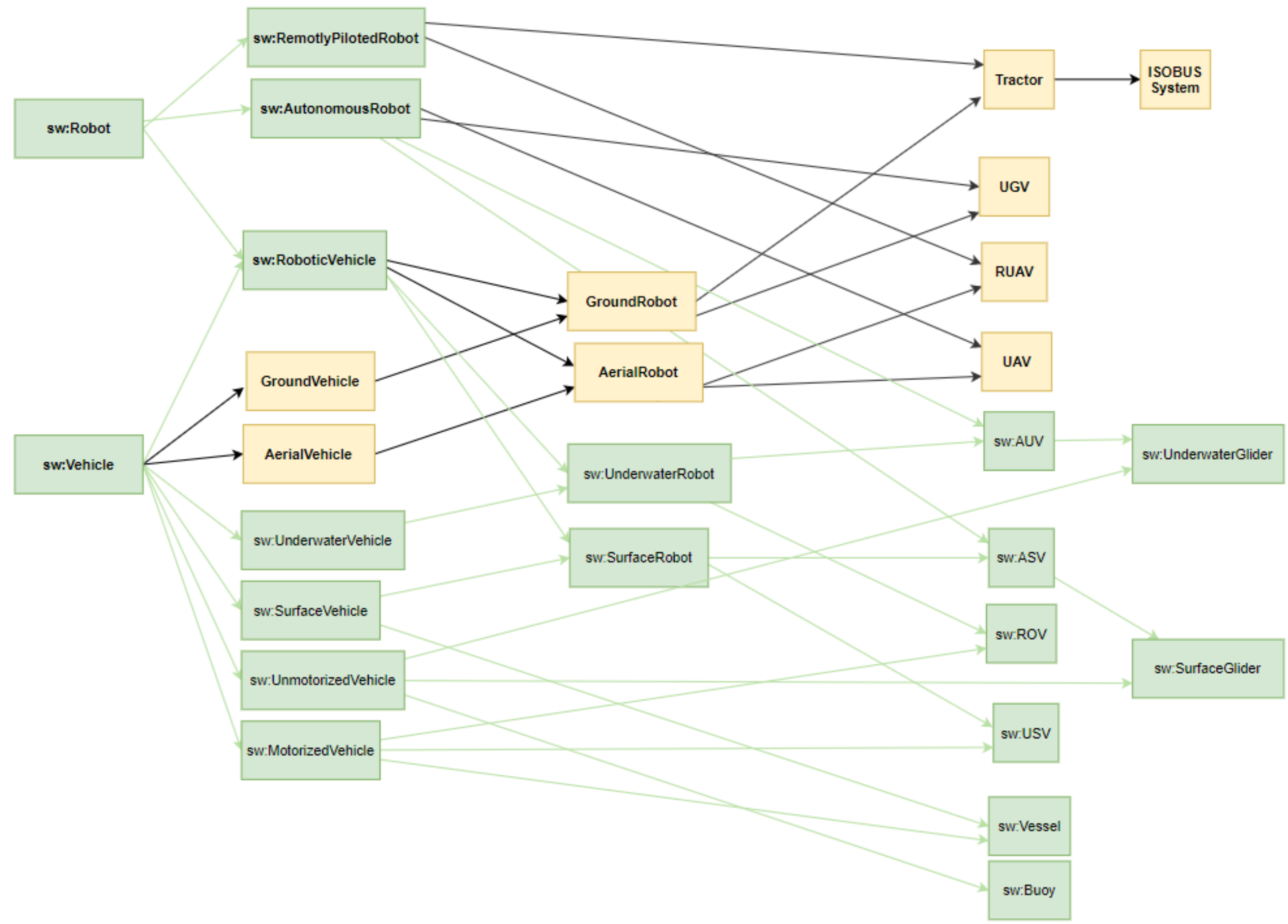

Figure 33. Hierarchy of AFarCloud vehicles

The Sensor domain is related to sensor data. It extends the Semantic Sensor Network (SSN) ontology [SSN17], and the Time ontology [TI20]. Figure 34 describes the main concepts defined in the Sensor domain.

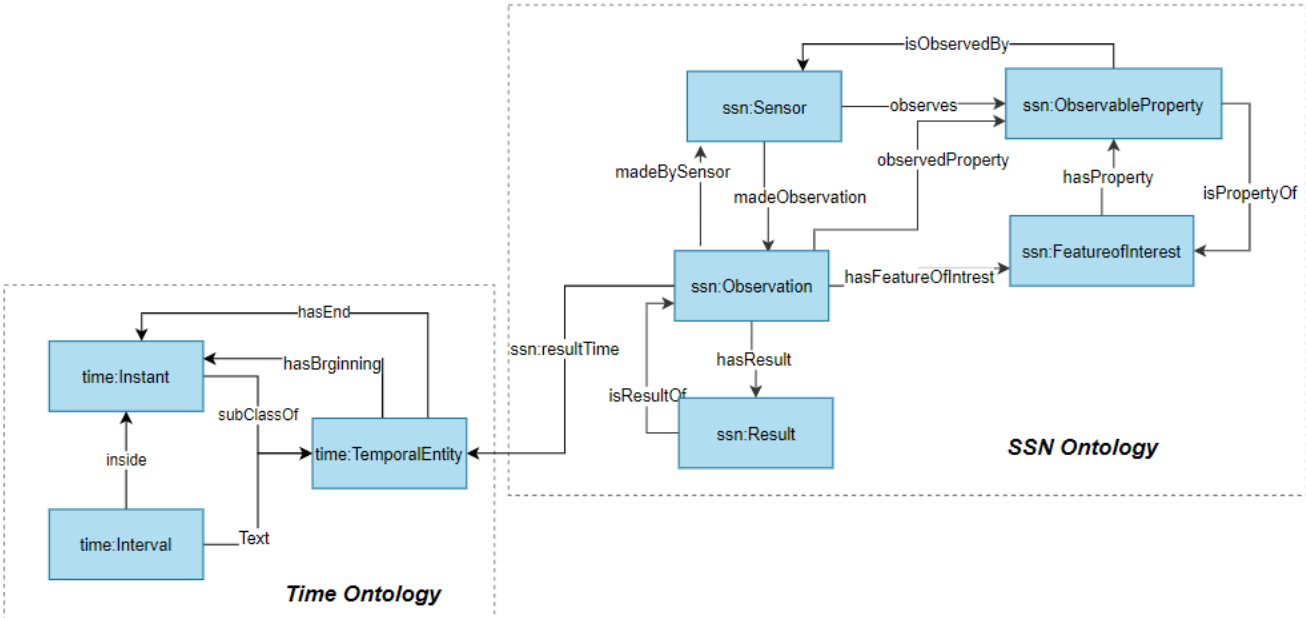

Figure 34. SSN main concepts and Time ontology used in the Sensor domain of AFarCloud

Finally, the Mission domain is responsible for providing a general representation of the mission composition and planning for UAV and UGV. This model provides a general representation of the mission goals, the mission plan, the tasks to be carried out by the vehicles and the required capabilities. A mission is defined as a set of goals to be performed by the vehicles. A goal is achieved by executing 1 to N tasks. These tasks can be of 3 types: *operator level*, *vehicle level* or *high-level* tasks. An *operator level task* is manually carried out by an operator. A *vehicle level task* can be carried out by one single vehicle, whereas a *high-level task* is an assembly of tasks that will be carried out by a swarm of vehicles. Tasks require capabilities to be performed, a minimum battery level and they have a start and end location. The main concepts defined in the Mission domain are described in Figure 35.

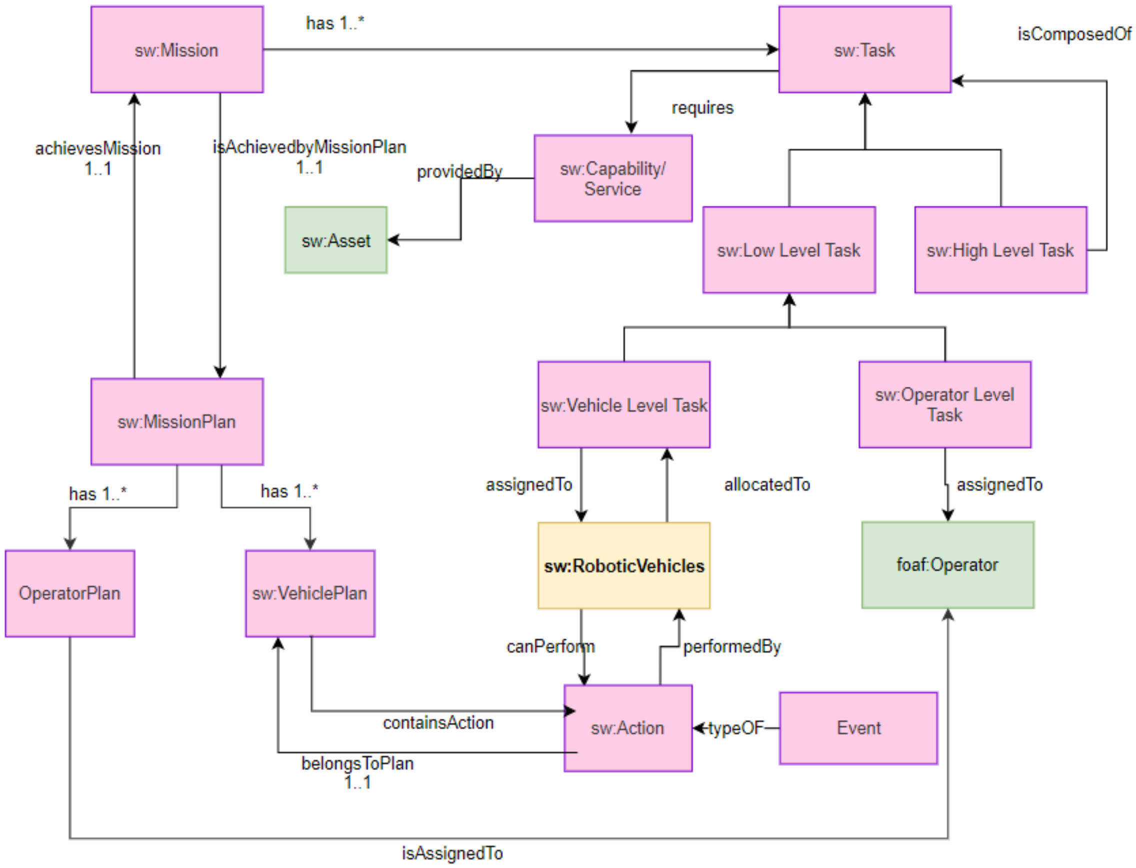

Figure 35. Structure of the Mission domain of AFarCloud

Impact and Incorporation of AFarCloud into the DEMETER AIM:

The AFarCloud project[45] aims to provide a distributed platform for autonomous farming that allows the AFarCloud ontology to be aligned with the DEMETER AIM in the following concepts classified by the type of data:

- Sensors, actuators, devices and observations (soil data and other sensor measurements): both ontologies are based on SSN and SOSA for modelling these concepts. Besides, AFarCloud extends the model by providing a list of 17 classes to represent the most common sensor types and 68 individuals of SOSA's ObservableProperty to represent the most common observation types used in agriculture.
  - Impact → DEMETER could reuse AFarCloud's vocabulary for sensor types and observation types in agriculture

[45] http://www.afarcloud.eu/

- Vehicle and mission related data: AFarCloud provides a model for representing mission planning with robotic vehicles. Examples of missions that are of interest for agriculture are: supervision of areas using collaborative UAVs, gathering observations from sensors using a UAV or UGV, generation of NDVI maps from images taken by UAVs, sending prescription maps to tractors, etc. This information could be used to extend the current AIM model.
    - Impact → DEMETER could be extended to cover concepts related to mission planning and hierarchy of robotic vehicles (UGV, AUV)
- Animals, dairy farms and crops: Both AFarCloud and DEMETER are aligned with the FOODIE project for modelling crops. Besides, AFarCloud provides a model for animals and dairy farms that could be useful for some pilots in DEMETER.
    - Impact → DEMETER could reuse concepts related to animals and dairy farms from AFarCloud.

In addition, as a result of the work in the Data Preparation & Integration enabler to transform the data format used in pilot 5.2 (based on the AFarCloud data model) to the DEMETER AIM, it was detected that the tool that was being used for the validation of the shapes, i.e., the TopBraid SHACL API, did not implement inference functionality. Hence, some models that were compliant to AIM were considered as not valid.

Observations in SOSA are related to a FeatureOfInterest. If you use a specific FeatureOfInterest such as AgriFarm, no inference is required. However, if you use an instance of Point which is a subclass of Geometry, the SHACL validator requires inference functionality.

For this reason, it is now recommended to use the tool PySHACL instead of the TopBraid SHACL API, whenever inference is required for the validation of the shapes.

Annex B: Questionnaire Responses

Table 12. Table with merged answers to AIM revision questionnaires

Type of AIM revision requested/ implemented*	Issue Reporting Date*	AIM Revision details*	Related Issue link (in WP2 Issue Tracker)	Related to Pilot X.Y or to Task x.y	Requested by*	Contact Person(s) *	Status of AIM revision	Other Comments
new extension	18.12.2020	Custom attributes for vehicle properties and driver behaviour	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/4	5.1, D2 component	DNET		concluded/ finalized	
new extension	18.12.2020	Custom fields for silos (volume, food type, density)	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/4	4.4 F2 component	DNET		concluded/ finalized	
new extension	18.12.2020	Custom fields for poultry well-being	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/4	4.4 component G2	DNET		concluded/ finalized	
new extension	18.12.2020	Custom fields for transport condition	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/4	5.4 5.1 H2 component	DNET		concluded/ finalized	

new extension	04/03/2021	If the project is being extended due to covid, then we could avail of that and extend the timeline for our tasks by say 6 months - as the lab work has been delayed due to covid and we have had difficulty in recruiting a post-doc		4.3	B. O'Brien		proposed	
change to Core Meta Model	04/03/2021	We had to change the pilot farm to one in UK - final discussions are taking place at the moment with this UK farm			B. O'Brien		concluded/ finalized	
change to Cross-domain ontology	29/01/2021	Some Crop missing properties (proposal). Related Agriculture ontologies: FIWARE AgriCrop, Agrifood; Saref4agri s4agri:Crop and s4agri:PlantGrowthStage.	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#8	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	

change to Cross-domain ontology	29/01/2021	Some Soil missing properties (proposal). Related Agriculture ontologies: FIWARE AgriParcelRecord; Saref4agri s4agri.	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#9	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	
change to Cross-domain ontology	29/01/2021	Some Irrigation missing properties and term (proposal). Related Agriculture ontologies: Saref4agri s4agri:WateringSystem	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#10	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	
change to Cross-domain ontology	29/01/2021	Some Weather missing properties (proposal). Related Agriculture datamodels: FIWARE WeatherObserved, Agrifood WeatherObserved.	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#11	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	
change to Cross-domain ontology	29/01/2021	Forecast missing term and properties (proposal). To represent forecasting results (irrigation, ETo, etc.)	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#12	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	

change to Cross-domain ontology	29/01/2021	Image missing term and properties (proposal). To represent an Image entity with an url to the image, etc.	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#13	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	
change to Cross-domain ontology	05/02/2021	Weather information time series (proposal). How to represent a timeseries of weather information (multiple attributes as wind speed, sun radiation, etc.)	demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies#14	Pilot 1.1_1.2	Manuel Mora (Universidad de Murcia)		pending	
change to Domain Specific ontologies	25/2/2021	Concept missing: Leaf Wetness (not humidity on the air) is missing from one of the following ontologies: agriCommon.ttl, agriCrop.ttl, agriProduct.ttl, agriProfile.ttl		Pilot 1.3	ICCS - George Routis	marios.paraskevopoulos@cn.ntua.gr	proposed	

new extension	25/2/2021	FertilizerEstimation extension with multiple data properties needed eg nitrogenUptake, nitrogenConcentration, soilRatio, chlorofyllIndex, mcari (chlorofyllAbsorption)	-	Pilot 1.3	ICCS - Marios Paraskevopoulos	marios.paraskevopoulos@cn.ntua.gr	proposed	would need more work to work out if any more concepts are also missing and need to be added. https://gitlab.com/demeterproject/wp2/dataanalytics/optimal-fertilizer-usage
change to Domain Specific ontologies	18/02/2021	Custom attributes for Plant Stress Detection	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/18	Pilot 1.4, Area A.3	SIMAVI Viorel Trusca		proposed	
change to Domain Specific ontologies	18/02/2021	Custom attributes for Nitrogen Balance Model	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/19	Pilot 1.4, Area C.1	SIMAVI Viorel Trusca		proposed	

new extension	04/03/2021	NOx in NOx in Sensor NOx out NOx out Sensor Charge Air Temperature DEF Level DEF Dosing Temperature DPF Regeneration Status Engine Coolant Temperature Fuel Temperature Engine Oil Temperature Engine Oil Pressure Crank Case Pressure Fuel Delivery Pressure Engine Coolant Pressure Engine Coolant Level Engine speed SCR in Temperature SCR out Temperature	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/21	Pilot 2.1	Andreas Schröder		proposed	

see the issue	see the issue	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/17	Diego Guiodtti	Diego Guiodtti			concluded/ finalized	An important required feature is to share a common list of AgriCrop. My "olive" should match with an other pilot "olive". I think that we coulddefine a set of demeter AgrICrop vocabolary to share among all pilots.

see the issue	see the issue	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/16	Diego Guiodtti	Diego Guiodtti			concluded/ finalized	I think that there is a misunderstanding on the AgriCrop element. There are two completely different concepts: crop: a reference to a plant species (e.g., Wheat-Triticum aestivum); parcel/field: an actual planttation with a crop, planted in a spcific place and time I am using AgriCrop for the crop and AgriParcel for the parcel/field In your comments: #3 (comment 505151129) You suggest to define a parcel and an harvest date inside an AgriCrop and I think that it is wrong. The parcel has a reference to a crop and not the opposite.

see the issue	see the issue	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/4	Diego Guiodtti	Diego Guiodtti			concluded/ finalized	IN the e.2 componnet I need to express for a specific location (expressed as a geo:Feature) a set of observation about the pest stage. The pest stage (it should be defined in http://www.demeter.org/ontology/prop#PestStage) has a numerical label (a code) and a description.
new extension	23/02/2021	Need to send images GeoPNG/GeoTiff (such NDVI, LAI...) Trap images inside JSON-LD/AIM using base64	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/1	Pilot 3.2	Filipe Santos		concluded/ finalized	This capacity will reduce the number of ports to be open and reduce authentication procedures.
alignment with existing ontology	11/02/2021	General reestructuring of the initial AIM model due to changes in component	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/15	Pilot 3.3	ATOS		pending	Pending meeting with Raul Palma

new extension	23/02/2021	extend Foodie:productionAmount to also include "predictedProductionAmount"	https://gitlab.com/demeterproject/wp2/issuetracker/-/issues/6	Pilot 3.4	Bart Beusen		proposed	instead of measured yield, this field should allow us to enter a value for the yield that is predicted by an algorithm
new extension	29/10/2020	The use of time series is recommended	N/D	Pilot 4.2	ENG	Massimo Giacalone (massimo.giacalone@eng.it)	concluded/ finalized	Issue link not defined as this extension was requested prior to using the Issue Traker
new extension	17/11/2020	Implemented property collections	N/D	Pilot 4.2	ENG	Massimo Giacalone (massimo.giacalone@eng.it)	concluded/ finalized	Issue link not defined as this extension was requested prior to using the Issue Traker
new extension	16/12/2020	Added new data properties to manage the prediction algorithm metrics	N/D	4.2	ENG	Massimo Giacalone (massimo.giacalone@eng.it)	concluded/ finalized	Issue link not defined as this extension was requested prior to using the Issue Traker The related software is available here[46]

[46] https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/blob/master/extensions/jsonld/livestockFeature-context.jsonld
https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/blob/master/extensions/livestockFeature.ttl
https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/blob/master/examples/example-timeseries-milkQualityPrediction-final.jsonld

alignment with existing ontology	24/11/2020	SHACL validation tool update to accept Point as a valid geometry of the FeatureOfInterest	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/2	Pilot5.2	TECNALIA + Sonia Bilbao		concluded/ finalized	Addressed by Raúl Palma related software is available here https://gitlab.com/demeterproject/wp5/pilots/pilot-5.2
new extension	10/12/2020	areaCultivated		Pilot 5.3	Ross Campbell		concluded/ finalized	
new extension	24/02/2021	meteoId	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/20	Pilot 5.3	Ross Campbell		pending	
new extension	10/12/2020	forecrops		Pilot 5.3	Ross Campbell		concluded/ finalized	
new extension	10/12/2020	seedAmount - added to DEMETER Ontology		Pilot 5.3	Ross Campbell		concluded/ finalized	

https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/blob/master/examples/example-timeseries-healthPrediction-final.jsonld

new extension	24/02/2021	plantDensity	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/20	Pilot 5.3	Ross Campbell		pending	
new extension	10/12/2020	soilResult - added to DEMETER ontology		Pilot 5.3	Ross Campbell		concluded/ finalized	
new extension	10/12/2020	news		Pilot 5.3	Ross Campbell		concluded/ finalized	
new extension	10/12/2020	issue		Pilot 5.3	Ross Campbell		concluded/ finalized	
new extension	08/01/2021	to provide input/output data for milk yield prediction API	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/5	4.F.1	Mimiro		proposed	Looks like we may need to have another extensions /model created to do with milk event data from NCDX. This would be to facilitate a data integration with NCDX.

change to Domain Specific ontologies	22/03/2021	KpiIndicator extension	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/23		DiegoGuidotti		proposed	I am planning to add to the Kpi a set of sub-sectors to better group togeher similar Indicators. I would like to maintain the sector object working as usuall add a sub-sector. For subsector i would use the SKOS ontology as done for the sectors
change to Domain Specific ontologies	27/03/2021	Data format	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/24		DiegoGuidotti		proposed	IN the following examples validFrom and validTo are nt is a standard yyyy-mm-dd data formats. It may be an issue during processing (we should enforce a valid date string for all date type) https://github.com/rapw3k/DEMETER/blob/master/models/examples/complex-farm-instance-AIM_v3.json-ld

change to Domain Specific ontologies	18 -03 - 2021	Time Series for String Status Device	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/22		Juan Andres Sanchez Segado		concluded/ finalized	I'm trying to define an observation time series for a string value.
change to Domain Specific ontologies	29-01-2021	AIM compliant time series proposal	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/7		Manuel Mora		concluded/ finalized	Here I attach a json-ld file with time series proposal for AIM v2 based in different examples found in: https://github.com/rapw3k/DEMETER/blob/master/models/examples/ We need to be sure this is right. This example has passed all AIM verification steps as indicated in issue "AIM compliant entities verification mandatory steps"

change to Domain Specific ontologies	11/12/2020	Fields not available in AIM	https://gitlab.com/demeterproject/wp2/agriculturalinformationmodel/domainspecificontologies/-/issues/3		Stefan Loureiro		concluded/ finalized	We are working on Area C, Component C2-Nutrient Monitor. We have created the AIM(JSON-LD) for our component, but some fields are not defined in AIM

Annex C: Detailed description of the AIM core meta-model

A meta-model, as its names implies, is a model of a model. Meta-models are typically used for different purposes. For instance, they can be used for the specification of modelling language constructs in a standardized, platform independent manner [HaPa09], to specify and restrict a domain in a data model and systems specification [IvVo11], or to provide an explicit model of the constructs and rules needed to build specific models within a domain of interest [Wel]. In fact, as noted in [Wel], meta-models can be viewed from three different perspectives: i) as a set of building blocks and rules used to build models; ii) as a model of a domain of interest; iii) as an instance of another model. In the context of the DEMETER meta-model, we are considering it as the first perspective.

Related to our context, it is important to highlight the relation between meta-models and ontologies. Ontologies provide shared vocabularies formally describing entities, such as concepts, properties and relations along with logical assertions (statements, rules), of a particular domain or that are common across multiple domains. They can be defined at different abstraction levels: top-level (foundation), domain, application. For instance, top-level ontologies define entities common across all/multiple domains and are the basis to support semantic interoperability among different domain-specific ontologies. Meta-models, on the other hand, are the explicit specification (constructs and rules) of how to build domain-specific models. They are targeted mainly at a structural specification of models, i.e., they are not intended to fully define their (logical) semantics [HaPa09]. Nevertheless, as stated in [Wel], a valid meta-model is an ontology, but ontologies may be defined at different abstraction levels, i.e., the meta-model may be regarded as an ontology used by modelers and integrated with ontologies at different abstraction levels. Thus, a foundational ontology (at the same abstraction level as a meta-model) and a domain ontology (at the same abstraction level as a (design) model) [HeSe11] can be integrated via an appropriate semantic mapping, which may be yet another ontology carrying a complementary set of semantics.

Based on the previous discussion of the use and benefits of meta-models, and after analysing different models and approaches (e.g., [HaPa09], [HeSe11], [SaKa07], [OMV14], [PaLi10]), we decided to follow the NGSI-LD meta-modeling approach [NGS1]. This approach is already a standard of the European Telecommunications Standards Institute (ETSI), whose mission is to make it easier for end-users, city databases, IoT and 3rd party applications to exchange information. Moreover, NGSI-LD is an evolution of the NGSI context interface family, particularly the FIWARE NGSI v2 information model, which was evolved by ETSI ISG CIM initiative to support linked data, property graphs and semantics, which is also a priority in DEMETER. Those standards were focused on the management of context information, which facilitate the development of smart solutions for different domains such as smart cities, smart industry, smart agrifood, and smart energy[47]. Here, context comprises all characteristics of all the entities (physical and nonphysical) involved in a target system/environment, as well as their states and other dynamic properties, together with relationships that stand for actual and virtual connections between them [ETS6]. Furthermore, the NGSI-LD meta-model provides a formal basis for

[47] https://www.fiware.org/developers/

representing "property graphs" using RDF/RDFS/OWL, making it possible to perform back and forth conversion between datasets based on the property graph model and linked data datasets that rely on RDF using blank-nodes reification. Moreover, the use of JSON-LD allows also the semantic referencing, where elements in the model can be matched to entities in well-known and/or standard ontologies. Following the NGSI-LD meta-modeling approach does not imply that we need to use either domain models such as FIWARE nor the domain model management methodology and tooling, however we can learn from, adopt, adapt and improve these aspects as necessary.

C.1 NGSI-LD overview

The NGSI-LD Information Model is defined at three levels. At the higher level, there are the foundation classes which correspond to the Core Meta-model and the Cross-Domain Ontology (see Figure 36 below). The former concerns the formal specification of the "property graph" model [Ro15]. The latter includes a set of generic, transversal classes which are aimed at avoiding conflicting or redundant definitions of the same classes in each of the domain-specific ontologies. Below these two levels, domain specific ontologies or vocabularies are devised.

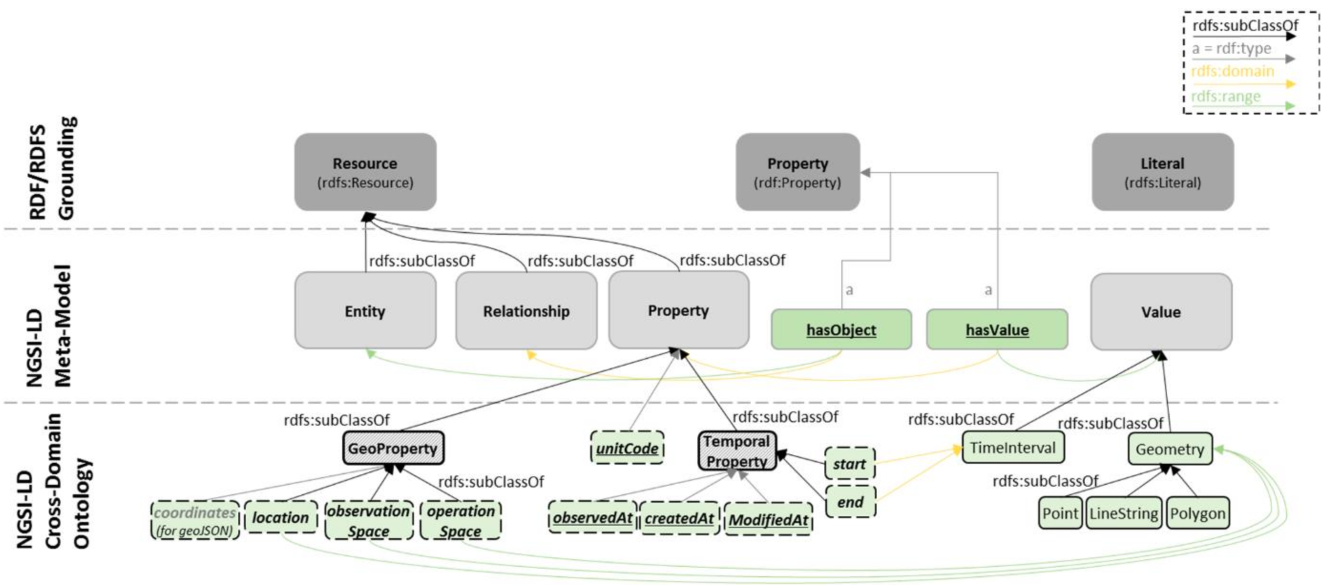

Figure 36. Meta-model and Cross-Domain Ontology High-Level View

NGSI-LD uses JSON-LD as main serialisation format, which provides the key advantage that terms can be defined in a separate document, referenced by an @context statement. In particular, the @context in JSON-LD is used to map terms provided as strings to concepts specified as URIs (ideally in ontologies).

NGSI-LD adopts a graph-based meta-model solution along with blank node reification, which is “especially convenient when the graph is serialized with JSON-LD because blank nodes do not explicitly appear in the textual serialized description, and actually show up only when it is represented as an RDF graph. It is thus possible for a developer to generate the JSON-LD payload of an API in a form that is very similar to what he would have generated in plain JSON [ETS6].

C.2 Separation of semantic referencing and structural descriptions

According to the specification [ETS6], NGSI-LD information model separates semantic referencing (as used in the Semantic Web) from the actual structural description. The structural description may be decomposed into a base structural graph whose nodes are physically matched entities, and an overlay layer that captures the way in which these entities are clustered into subgraphs.

The semantic referencing in NGSI-LD is in theory based on standard RDF/RDS/OWL typing and public ontologies. Accordingly, all nodes and edges of the structural graph are matched to several relevant classes/categories of such ontologies, which together characterize the features shared by all the instances of these classes.

The structural graph is as a model of the structural description of an environment and captures the relationships between the different subsystems that make up this environment. This is, according to the specification, to some extent independent of the overlaying semantic referencing and it could be considered to "stand on its own", even without such referencing.

In contrast to the semantics “per resource” that RDF is meant to describe (e.g., via referencing), the structural graph has a different kind of semantics of its own that apply to the graph as a whole, such as e.g. when a graph captures and matches the structure of a physical network like a power grid or a water distribution network.

In the implementation, though, the semantic referencing mentioned here is not really followed, as discussed later in the insights.

C.3 NGSI-LD meta model

According to the specification [ETS6], the NGSI-LD meta-model provides a formal basis for representing "property graphs" using RDF/RDFS/OWL. This makes it possible to perform back and forth conversion between datasets based on the property graph model and linked data datasets that declare more formal semantics using RDF. This could be described as raising the semantic expressivity of RDF triples to the level of property graphs, as for instance, property graphs may use predicates as subjects of other predicates (properties of properties and properties of relationships). Conversely, it may be described as grounding the semantics of property graph elements in discoverable definitions and using this to constrain arbitrary and non-interoperable proliferation of similar property graph patterns for the many specific cases that need to be modelled.

A graph-based model was chosen as it enables to capture the complex structure and inter-entity relationships describing the characteristics of entities (physical and non-physical) involved in systems for smart solutions, which make up the context information. Such information may be natively structural information only, with no semantic definitions from the beginning; the semantics of this context may be added in a later stage of graph enrichment. This is the case of the current implementation of the NGSI-LD context, where semantics are not defined yet (see discussion below). A more detailed discussion about the NGSI-LD context can be found in D2.1 Section 7.1.4.

C.4 NGSI-LD summary

NGSI-LD approach is well founded, following a layered architecture and based on the increasingly popular JSON-LD serialisation format. Conceptually, it enables the good sides of two “worlds”: the benefits of linked data and underlying RDF-based reasoning tools and querying (enabling data integration, knowledge discovery, etc.), and the richer expressivity of property graphs (using predicates as subjects of other predicates).
The current challenges we foresee, which could also be used as feedback for future developments in that community, are more on the implementation of this approach:

- The current NGSI-LD context *is a simple flat schema* that includes the meta-model and cross ontology terms without any explicit semantics. Except from some property JSON types (@type: DateTime, id), there are no definition that a term is a class, a property with explicit information about the type of property (e.g., relation, datatype), constraints on domains/ranges, cardinality, taxonomic relations, or other axioms. Of course, the JSON @type would allow to infer that a given term is a relation (@type: @id), but even those with @type: DateTime are not defined explicitly with the type of property it is, as DateTime (https://uri.etsi.org/ngsi-ld/DateTime) is not having any explicit semantic information.
- The terms *are not mapped to any standard and/or well-known ontologies/vocabularies (no reuse)*. NGSI-LD specification discusses such approach via the semantic referencing, but the context implementation is not including them; perhaps they are considered *to be added in a later stage* (as also mentioned in the documentation). There is also available documentation (see Annex B of [ETS6]) discussing mappings to some well-known ontologies/vocabularies (such as oneM2M, W3C WoT Thing Description, W3C Time Ontology and SAREF); however, no implementation seems to be available to allow any integration. In fact, it is not clear, how such mappings would be implemented from the documentation reviewed.
- *Other modules/profiles (domain vocabularies) are defined in the same way*, i.e., simple flat schemas with no mapping/reuse of existing standards and/or well-known ontologies. For instance, FIWARE Data Models @context[48] is used in many of the provided examples and is part of the full @context[49] (which

[48] https://fiware.github.io/data-models/context.jsonld
[49] https://fiware.github.io/data-models/full-context.jsonld

also includes the core @context) of NGSI-LD. FIWARE @context defines many entities related to different FIWARE related domains. The full list set of models, called the FIWARE Smart Data Models[50], provide different json schemas and data examples in json and json-ld. The project GSMA IoT also provides a repository of different NGSI-LD entities from different domains[51], more complete than FIWARE, with a specification, json-ld context example and data example in json-ld. An important difference, though, is that GSMA IoT includes some references to well-known ontologies or vocabularies. However, such references are very few and the context are only samples. Most of the terms are still defined ad-hoc.

- The flat schema implementation approach is not scalable, and difficult to maintain.
- The only semantic information available is in fact included in the encoding of data itself, and it is provided by the meta-model (e.g., an element is a property or a relationship). For instance, the encoding of a FIWARE agri-parcel entity is (partially) below (the full encoding of the example is also available for download[52]).
- There is sufficient complexity and evidence for the benefits for adaptation of tools to manage, collate, validate and document the DEMETER AIM using a similar approach to FIWARE, but with extended capabilities as required: for example, to create and exploit more interoperable intermediate profiles.


```
{ "@context": [
     "https://schema.lab.fiware.org/ld/context.jsonld",
     "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
   ],
      "@id": "urn:demeter:silos:5812fbbe-3ce5-4b81-a74a-b680000a5bef",
    "@type": "AgriParcelRecord",
      "createdAt": "2017-01-01T01:20:00Z",
      "modifiedAt": "2017-05-04T12:30:00Z",
      "source": "https://source.example.com",
      "dataProvider": "https://provide.example.com",
      "entityVersion": "2.0",
      "hasAgriParcel": {
        "type": "Relationship",
```


50 https://github.com/smart-data-models/data-models
51 https://github.com/GSMADeveloper/NGSI-LD-Entities
52 https://github.com/rapw3k/DEMETER/blob/master/ngsi-ld-fiware-parcel-example.jsonld

```
        "object": "urn:ngsi-ld:AgriParcel:d3676010-d815-468c-9e01-25739c5a25ed"
      },
      "soilTemperature": {
        "type": "Property",
        "value": 27,
        "unitCode": "CEL",
        "observedAt": "2017-05-04T12:30:00Z"
      },
      "observedAt": {
        "type": "Property",
        "value": "2017-05-04T10:18:16Z"
      }
}
```

The transformation of that json-ld into RDF would be as follows:

```
@prefix fiware: <https://smart-data-models.github.io/data-
models/terms.jsonld#/definitions/> .
@prefix ngsi-ld: <https://uri.etsi.org/ngsi-ld/> .
@prefix ngsi-ld-default: <https://uri.etsi.org/ngsi-ld/default-context/> .

<urn:demeter:silos:5812fbbe-3ce5-4b81-a74a-b680000a5bef>
  a <https://uri.fiware.org/ns/data-models#AgriParcelRecord> ;
  fiware:dataProvider "https://provide.example.com" ;
  fiware:hasAgriParcel [
    a <https://uri.etsi.org/ngsi-ld/Relationship> ;
    ngsi-ld:hasObject <urn:ngsi-ld:AgriParcel:d3676010-d815-468c-9e01-
25739c5a25ed>
  ] ;
  fiware:soilTemperature [
    a ngsi-ld:Property ;
```

```
    ngsi-ld:hasValue 27 ;
    ngsi-ld:observedAt "2017-05-04T12:30:00Z"^^ngsi-ld:DateTime ;
    ngsi-ld:unitCode "CEL"
  ] ;
  fiware:source "https://source.example.com" ;
  ngsi-ld:createdAt "2017-01-01T01:20:00Z"^^ngsi-ld:DateTime ;
  ngsi-ld-default:entityVersion "2.0" ;
  ngsi-ld:modifiedAt "2017-05-04T12:30:00Z"^^ngsi-ld:DateTime ;
  ngsi-ld:observedAt [
    a ngsi-ld:Property ;
    ngsi-ld:hasValue "2017-05-04T10:18:16Z"
  ] .
```

As the FIWARE @context does not link to any ontology, but the entities are defined ad-hoc, there are no explicit semantics. As a result, many advantages of the linked data and underlying RDF-based reasoning tools and querying cannot be easily or directly exploited, e.g., (automatic) data link discovery (integration), (automatic) model alignment for data integration, validation of conformance of data with the model with a simple reasoner, inferencing on the data to discover new knowledge, specialisations (taxonomy) with inheritance of axioms.

C.5 DEMETER AIM cross-domain considerations

Our approach for the design of DEMETER AIM, discussed at the beginning of this section, is similar and in line with the NGSI-LD approach, i.e., modular in a layered architecture. Our first design choice, though, was to decide whether to follow a 2-layer approach (top-level/cross domain + domain ontologies) with direct grounding on RDF/RDFS/OWL or a n-layer approach, as in NGSI-LD (where n=3), which includes the property graph meta-model layer (grounded on RDF/RDFS) at the highest level. After further analysis of the NGSI-LD specification, we decided on the latter for the following reasons:

1. It allows DEMETER AIM to be compliant and easily integrated with NGSI-LD data and models, thus facilitating the integration with existing datasets based on these models that may be relevant to DEMETER.
2. It allows natively the representation of the rich and complex context information of different entities (e.g., systems/platforms/environments) typical within IoT (or WoT) applications, where the context

includes the set of properties characterizing these entities, together with the set of relationships that enmesh them together, and the properties of these relationships and properties. This was the main motivation of NGSI-LD and it is also a very important aspect for DEMETER.

3. It allows having the best of two "worlds": property-graphs and linked data. It allows to perform back and forth conversion between datasets based on the property graph and linked data datasets that rely on the RDF framework. As described in [ETS6], property graphs are the implicit semi-formal data models underlying most present-day graph databases, which have gained widespread use especially in the industry (as opposed to academia). They make it possible to attach properties (defined as key-value pairs) to relationships and other properties, a feature which RDF does not directly support, but they lack the standardization and formal underpinnings of RDF and do not interoperate directly with linked data and other RDF datasets. Also, they do not lend themselves to reasoning with RDF-based reasoning tools or querying with standard query languages such as SPARQL.

Thus, DEMETER AIM initially followed the same 3-layer architecture of NGSI-LD, including a property graph meta-model layer (grounded in RDF/RDFS), a cross-domain ontologies layer, and the domain/application ontologies. This was then extended with an additional layer comprising DEMETER's pilot specific extensions. However, as opposed to NGSI-LD, DEMETER AIM implements the cross-domain and domain/application layers by reusing existing standards and/or well-known ontologies/vocabularies as much as possible from the outset, thereby implementing semantic referencing. As an example, consider the following agriculture management zone using FOODIE ontology[53] (one of the base ontologies used for the domain layer as discussed later in Section 9.3) as the underlying model encoded in RDF/turtle.

```
@prefix foodie: <http://foodie-cloud.com/model/foodie#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix fiware: <https://uri.fiware.org/ns/data-models#> .

<http://w3id.org/foodie/core/ManagementZone/4>
  a <http://foodie-cloud.com/model/foodie#ManagementZone> ;
  foodie:code "CODA4"^^xsd:string ;
  fiware:entityVersion 2 ;

  foodie:creationDateTime "2015-12-01T00:00:00"^^xsd:dateTime ;
```

[53] http://agroportal.lirmm.fr/ontologies/FOODIE

```
  foodie:cropSpecies <http://w3id.org/foodie/core/CropType/20> ;
  foodie:holdingZone <http://w3id.org/foodie/core/Plot/1> ;
  foodie:originType <http://w3id.org/foodie/core/OriginTypeValue/1> ;
  foodie:zoneAlert <http://w3id.org/foodie/core/Alert/4> ;
  geo:hasGeometry <http://w3id.org/foodie/core/ManagementZone/4/geometry> ;
  rdfs:label "ManagementZone #4"^^xsd:string .
```

With DEMETER AIM, we would define an agriculture model module/profile as a JSON-LD @context, which defines the terms used in DEMETER by reusing existing standards and/or well-known ontologies/vocabularies, such as Saref4Agri or FOODIE, i.e., mapping DEMETER terms to the reused ontology/vocabulary terms. A partial example of such @context[54] (using FOODIE ontology elements for demonstration purposes) would be as follows.

```
{
"@context": {
        "xsd" : "http://www.w3.org/2001/XMLSchema#",
        "Nutrients": "http://foodie-cloud.com/model/foodie#ProductNutrients",
        "Plot": "http://foodie-cloud.com/model/foodie#Plot",
        "DoseUnit": "http://foodie-cloud.com/model/foodie#DoseUnit",
        "TreatmentPlan": "http://foodie-cloud.com/model/foodie#TreatmentPlan",
        "ManagementZone": "http://foodie-cloud.com/model/foodie#ManagementZone",
        "Intervention" : "http://foodie-cloud.com/model/foodie#Intervention",
        "CropSpecies" : "http://foodie-cloud.com/model/foodie#CropSpecies",
        "Treatment" : "http://foodie-cloud.com/model/foodie#Treatment",
        "Holding" : "http://inspire.ec.europa.eu/schemas/af/3.0#Holding",
        "code" : "http://foodie-cloud.com/model/foodie#code",
        "creationDateTime" : {
                        "@id" : "http://foodie-
cloud.com/model/foodie#creationDateTime",
                        "@type": "xsd:dateTime"
```

54 https://github.com/rapw3k/DEMETER/blob/master/DEMETER-agricontext.jsonld

```
        },
        "cropSpecies" : {
                        "@id" : "http://foodie-cloud.com/model/foodie#cropSpecies",
                        "@type": "@id"
        },
        "originType" : {
                        "@id" : "http://foodie-cloud.com/model/foodie#originType",
                        "@type": "@id"
        },
        "zoneAlert" : {
                        "@id" : "http://foodie-cloud.com/model/foodie#zoneAlert",
                        "@type": "@id"
        },
        "holdingZone" : {
                        "@id" : "http://foodie-cloud.com/model/foodie#holdingZone",
                        "@type": "@id"
        }
}}
```

Then, the encoding of the same management zone presented above in JSON-LD using DEMETER AIM would look like the listing below (also available for download[55]), which could be easily transformed back to RDF[56] to get the same listing as above. Note that in addition to the agriculture context, we are adding two more terms to the context in this example (namely: label and geometry); however, such terms would be defined in the different profiles/modules at the cross-domain level (e.g., geospatial model) (see cross-domain layer section).

```
{
"@context": [
          "https://rapw3k.github.io/DEMETER/DEMETER-agricontext.jsonld",
```

[55] https://github.com/rapw3k/DEMETER/blob/master/managementZone4-example.jsonld
[56] http://www.easyrdf.org/converter

```
          {"label": "http://www.w3.org/2000/01/rdf-schema#label",
           "geometry": {
               "@id": "http://www.opengis.net/ont/geosparql#hasGeometry",
               "@type":"@id"
              }
          }
     ],
         "@id": "http://w3id.org/foodie/core/ManagementZone/4",
         "@type": "ManagementZone",
         "label": "ManagementZone #4",
         "code": "COD4",
         "creationDateTime" : "2015-12-01T00:00:00",
         "cropSpecies" : "http://w3id.org/foodie/core/Croptype/20",
         "holdingZone" : "http://w3id.org/foodie/core/Plot/1",
         "originType" : "http://w3id.org/foodie/core/OriginTypeValue/1",
         "zoneAlert" : "http://w3id.org/foodie/core/Alert/4",
         "geometry" : "http://w3id.org/foodie/core/ManagementZone/4/geometry"
}
```

Note, however, that if we would like to use the expressivity of the property graph model (to raise the semantic expressivity of RDF triples to the level of property graphs), we would first define our core meta-model @context[57] (same as for NGSI-LD) as the listing below:

```
 "@context": {
      "id": "@id",
      "type": "@type",
      "value": "https://uri.etsi.org/ngsi-ld/hasValue",
      "object": {
        "@id": "https://uri.etsi.org/ngsi-ld/hasObject",
```

[57] https://github.com/rapw3k/DEMETER/blob/master/DEMETER-core-meta-model.jsonld

```
          "@type": "@id"
        },
        "Property": "https://uri.etsi.org/ngsi-ld/Property",
        "Relationship": "https://uri.etsi.org/ngsi-ld/Relationship"
}
```

Then we would be able to attach properties to relationships or other properties (i.e., using the property graph model). So, in our previous example, if we would like to attach properties to one of our data type properties (e.g., code to include for instance the codelist name or organisation name giving such code), and to one object property (e.g., cropSpecies to say for instance at what time this information was captured), the encoding of the previous management zone would be as the listing below (also available for download[58]). Note that no extra properties are attached in the example though, as this is just for illustration).

```
{
 "@context": [
    "https://rapw3k.github.io/DEMETER/DEMETER-agricontext.jsonld",
    "https://rapw3k.github.io/DEMETER/DEMETER-core-metamodel.jsonld",
    {
     "label" : "http://www.w3.org/2000/01/rdf-schema#label",
     "geometry": {
            "@id": "http://www.opengis.net/ont/geosparql#hasGeometry",
            "@type":"@id"
            }
     }
   ],
   "id": "http://w3id.org/foodie/core/ManagementZone/4",
   "type": "ManagementZone",
   "label": "ManagementZone #4",
   "code": {
```

[58] https://github.com/rapw3k/DEMETER/blob/master/managementZone4-example-property-graph.jsonld

```
      "type": "Property",
      "value": "CODA4"
    },
   "creationDateTime" : "2015-12-01T00:00:00",
   "cropSpecies" : {
       "type": "Relationship",
       "object": "http://w3id.org/foodie/core/CropType/20"
   },
   "holdingZone" : "http://w3id.org/foodie/core/Plot/1",
   "originType" : "http://w3id.org/foodie/core/OriginTypeValue/1",
   "zoneAlert" : "http://w3id.org/foodie/core/Alert/4",
   "geometry" : "http://w3id.org/foodie/core/ManagementZone/4/geometry"
}
```

Now, if we see the corresponding RDF/Turtle representation, it would be like the listing below:

```
@prefix foodie: <http://foodie-cloud.com/model/foodie#> .
@prefix ngsi-ld: <https://uri.etsi.org/ngsi-ld/> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
<http://w3id.org/foodie/core/ManagementZone/4>
  a foodie:ManagementZone ;
  foodie:code [
    a ngsi-ld:Property ;
    ngsi-ld:hasValue "CODA4"^^xsd:string
  ] ;
  foodie:creationDateTime "2015-12-01T00:00:00"^^xsd:dateTime ;
  foodie:cropSpecies [
    a ngsi-ld:Relationship ;
```

```
    ngsi-ld:hasObject <http://w3id.org/foodie/core/CropType/20>
  ] ;
  foodie:holdingZone <http://w3id.org/foodie/core/Plot/1> ;
  foodie:originType <http://w3id.org/foodie/core/OriginTypeValue/1> ;
  foodie:zoneAlert <http://w3id.org/foodie/core/Alert/4> ;
  geo:hasGeometry <http://w3id.org/foodie/core/ManagementZone/4/geometry> ;
  rdfs:label "ManagementZone #4"^^xsd:string .
```

That last individual, however, is not a valid OWL 2 DL definition, though, as properties are used differently from how they were defined in the referenced (re-used) ontology (FOODIE). In particular, the data type property *code* is used as an annotation/object property pointing to a blank node of type *Property*, and object property *cropSpecies* defined with range *CropType* is used as an annotation/object property pointing to a blank node of type *Relationship*. OWL 2 DL allows punning[59] allowing to use, e.g., an URI as class and as an individual, but still has limitations, i.e., a name cannot be used for both a class and a datatype and a name can only be used for one kind of property. Hence such individual would be treated as an RDF graph (OWL 2 Full). If we would like to move back to OWL DL to make use of, e.g., reasoning, we would need to convert back to lower expressivity (remove the property graphs).

Finally, it is worth noting that, similar as the NGSI-LD approach presented in annex B of [ETS6] mentioned above, DEMETER AIM will map entities from selected major ontologies/vocabularies for the cross-domain to the core AIM meta-model.

[59] https://www.w3.org/TR/owl2-new-features/#F12:_Punning

Annex D: Detailed description of the AIM Domain-Specific ontologies

In this section, the complete DEMETER AIM domain-specific layer is presented, including unchanged and updated modules.

D.1 Agriculture Profile ontology

The Agriculture Profile ontology is available under https://w3id.org/demeter/agri. This imports all the remaining ontologies used in AIM and is briefly presented below.

```
@prefix: < https://w3id.org/cybele/> .
@prefix qb: < http://purl.org/linked-data/cube#> .
@prefix dct: < http://purl.org/dc/terms/> .
@prefix owl: < http://www.w3.org/2002/07/owl#> .
@prefix rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: < http://www.w3.org/XML/1998/namespace> .
@prefix xsd: < http://www.w3.org/2001/XMLSchema#> .
@prefix dcat: < http://www.w3.org/ns/dcat#> .[60]
@prefix foaf: < http://xmlns.com/foaf/0.1/> .[61]
@prefix prov: < http://www.w3.org/ns/prov#> .
@prefix rdfs: < http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: < http://www.w3.org/2004/02/skos/core#> .
@prefix stat: < http://data.europa.eu/m8g/> .
@prefix schema: < http://schema.org/> .
@prefix af - inspire: < http://inspire.ec.europa.eu/schemas/af/3.0#> .
@prefix act - inspire: < http://inspire.ec.europa.eu/schemas/act-core/3.0#> .
@prefix foodie: < http://foodie-cloud.com/model/foodie#> .
@prefix saref4agri: < https://w3id.org/def/saref4agri#> .
@prefix common: < http://portele.de/ont/inspire/baseInspire#> .
@prefix fiware: < https://uri.fiware.org/ns/data-models#> .
@prefix iso19109: < http://def.seegrid.csiro.au/isotc211/iso19109/2005/feature#> .
@prefix iso19150 - 2: < http://def.seegrid.csiro.au/isotc211/iso19150/-
2/2012/basic#> .
@prefix iso19103: < http://def.seegrid.csiro.au/isotc211/iso19103/2005/basic#> .
@prefix geo: < http://www.opengis.net/ont/geosparql#> .
@prefix saref: < https://w3id.org/saref#> .
@prefix ssn: < http://www.w3.org/ns/ssn/> .
@prefix obo: < http://purl.obolibrary.org/obo/> .
@base < https://w3id.org/demeter/> .
```

[60] https://www.w3.org/TR/vocab-dcat-2/
[61] https://en.wikipedia.org/wiki/FOAF_(ontology)

```
< https://w3id.org/demeter/agri> rdf:type owl:Ontology ;
      owl: versionIRI <
https://raw.githubusercontent.com/rapw3k/DEMETER/master/models/demeterAgriProfile.t
tl> ;
      dct: contributor[schema:affiliation[foaf:name "OGC"];
                      foaf:name "Rob Atkinson"],
                      [schema:affiliation[foaf:name "ICCS"];
                          foaf:name "Ioanna Roussaki"];
      dct:creator[schema:affiliation[foaf:name "PSNC"];
                       rdfs:seeAlso < http://orcid.org/0000-0003-4289-4922> ;
                       foaf: name "Raul Palma"];
      owl:imports < https://w3id.org/demeter/agri/agriCommon> ,
                       < https://w3id.org/demeter/agri/agriFeature> ,
                        < https://w3id.org/demeter/agri/agriCrop> ,
                        < https://w3id.org/demeter/agri/agriIntervention> ,
                        < https://w3id.org/demeter/agri/agriAlert> ,
                        < https://w3id.org/demeter/agri/agriProduct> ,
                        < https://w3id.org/demeter/agri/agriProperty> ,
                        < https://w3id.org/demeter/agri/agriSystem> ,
                        < https://w3id.org/demeter/agri/agriPest> ,
                        < https://w3id.org/demeter/agri/farmAnimal> ,
                        < https://w3id.org/demeter/agri/agriResource> ;
       dct: description "The DEMETER Agri Profile is a master profile importing
focused specific profiles/modules of DEMETER AIM."@en;
       dct: rights "This vocabulary is distributed under Creative Commons
Attribution 4.0 License - http://creativecommons.org/licenses/by/4.0"@en;
       dct: title "DEMETER AgriCrop"@en;
       rdfs: comment "The DEMETER Agriculture Information Model (AIM) is the common
vocabulary in DEMETER project providing the basis for semantic interoperability
across smart farming solutions"@en;
       owl: versionInfo "1.0";
      foaf: maker[foaf:homepage < https://h2020-demeter.eu/> ;
                  foaf: name "DEMETER project"] .
```

As can be seen in the previous depicted .ttl file, it starts with prefixes *@prefix* allowing to declare instead of a long prefix of the repeated URI with a short prefix. In the .ttl[62] file, you can also see the *@base <https://w3id.org/demeter/>* . The *@base* allows extra abbreviation of URIs, however it is often used for simplifying the URIs in the data, where the prefix directives are for vocabularies which describe the data. You can also see that this also uses the *foaf* ontology (*friend of a friend)* which is a machine-readable ontology describing persons, their activities and how they are related to other people and objects. It also defines

[62] https://www.w3.org/TeamSubmission/turtle/

prefixes for a number of other ontologies that are being used, including DCAT (used by the metadata schema presented in section 7.4), FOODIE, saref4agri, FIWARE and SSN.

The demeterAgriProfile ontology imports (and therefore consists) of the following ontologies: *agriCommon, agriFeature, agriCrop, agriIntervention, agriAlert, agriProduct, agriProperty, agriSystem, agriPest, farmAnimal.* The initial/core classes are the following: *ActivityComplex, Agent, Agri Farm, Agri Parcel, AgriParcelOperation, AgriParcelRecord, AgriPest, AgriProductType, Alert, Animal, AnyFeature, Codelist, Datatype, Deployment, EconomicActivityNACEValue, Feature of interest, FeatureType, ID, Measure, Measurement, Period, Platform, Property, skos:Concept, SpatialObject, System, taxonomic_rank, Unit of measure.*

D.2 Agriculture Commons ontology

The Agriculture Commons ontology is available under https://w3id.org/demeter/agri/agriCommon and is presented in the figure below.

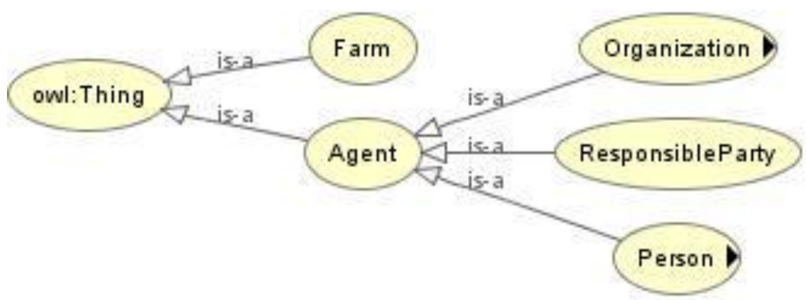

Figure 37. Visualization of the Agriculture Commons ontology

In addition, a few basic classes and concepts are defined in this module: *Agent* and its subclass *Person* are taken from the FOAF ontology and a subclass *Farmer* is defined as well for use in the Demeter ontology. In the same manner, the general class *Organization* is subclassed by the *FarmHolding* class.

The classes *Agent, Person, Organization* and *Role* have been elevated to the cross-domain layer.

D.3 Agriculture Features ontology

The Agriculture Features ontology is available under https://w3id.org/demeter/agri/agriFeature and is briefly presented below.

The agriFeature.ttl consists of the following classes: *ActivityComplex* class, with the following subclasses: *Farm* class (defined by Saref4agri) which defines a plot of land that is used for the scope of farming, containing buildings and parcels, and *Holding* class. *Agri Farm* class (defined by Fiware) which describes a generic farm constituting of buildings and parcels. *Agri Parce*l which describes the conditions recorded in a generic greenhouse, with subclass *Agri Greenhouse*. *AnyFeature* class, with subclasses: *Farm* class (defined by Saref4agri) which is used for farming containing buildings and parcels, *Holding* class, *ManagementZone* class (defined by Foodie), *Parcel* class (defined by Saref4agri) which is an area of land that cannot be divided and contains homogeneous items, *Plot* class (defined by Foodie) and *Site* class. *Codelist* (defined by Foodie) class, with *OriginTypeValue* (defined by Foodie). *Crop* class (defined by Saref4agri), *EconomicActivityNACEValue* class, *FeatureType* class, with subclasses: *Farm*, *Holding*, *ManagementZone*, *Parcel*, *Plot* and *Site*. *MachineType* class, *PropertyType* class, *skos:Concept* class with subclass: *OriginTypeValue* (defined by Foodie). *SpatialObject* class which represents everything that can have a spatial representation, with subclasses: *Feature* which represents the top-level feature type and it is similar to GFI_Feature of ISO 19156:2011, and subclasses: *Agri Farm* class, *Agri Parcel* class, (with subclass *Agri Greenhouse*), *Building* class (defined by Saref4agri) which represents a structure providing shelter for its occupants or contents and stands in one place, *Building space* class (defined by Saref4agri) which is used to define the physical spaces of the building, *Farm*, *Holding*, *ManagementZone*, *Parcel*, *Plot* and *Site*. The last class is *TractorType*.

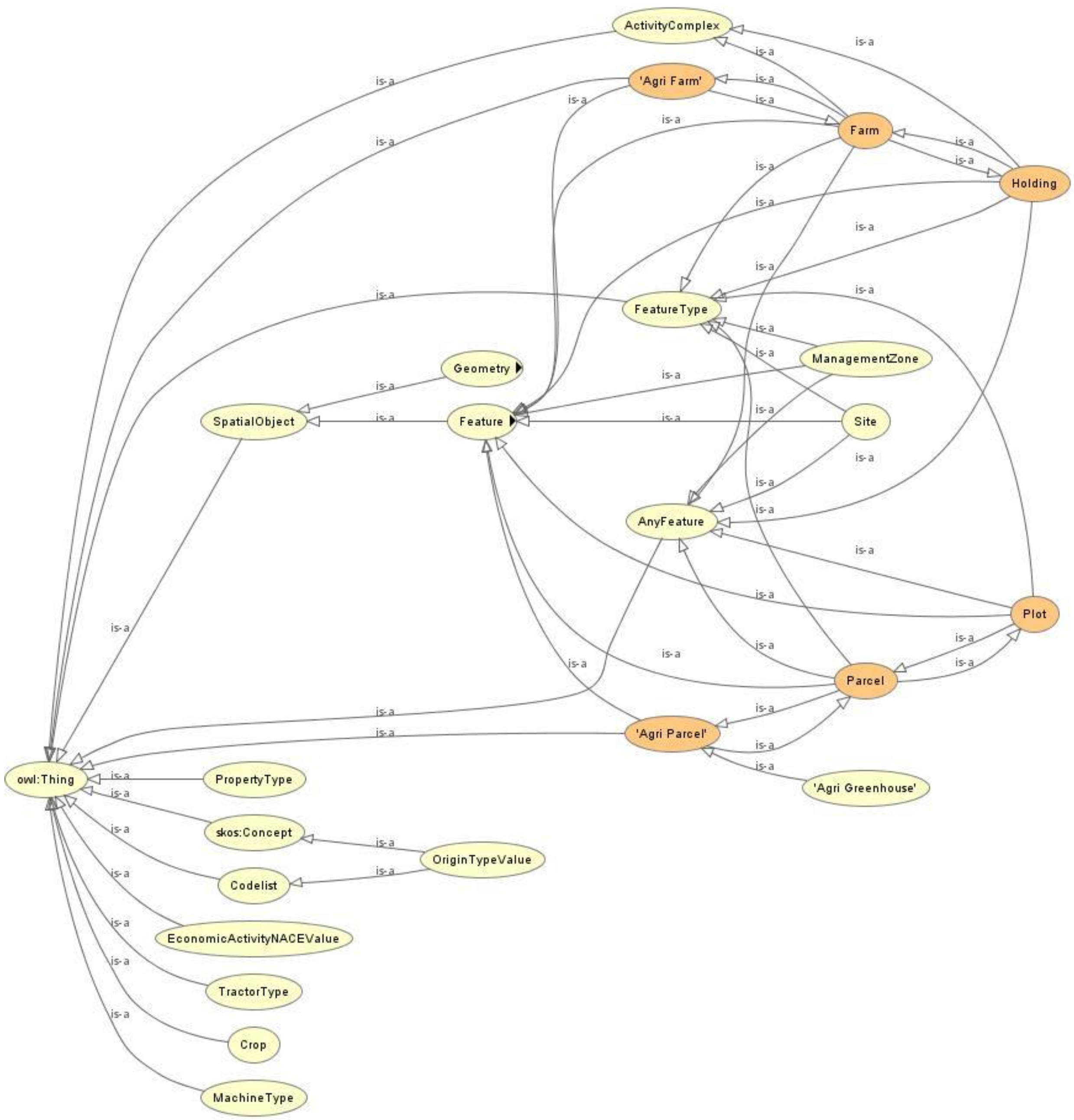

Figure 38. Visualization of the Agriculture Features ontology

Concerning the Object properties of this ontology, the following are used: *Activity*, *Contains* with subclasses: "Contains" with subclasses: *contains*, *containsPlot*, *containsZone*, *hasAgriParcel*, *hasAgriParcelChildren*. *Crop property*, *geo:location*, *has geometry* with subclasses: *landLocation*"and *Location*. *hasAgriCrop* property, *hasAgriSoil*, *hasDevice*, *includesAnimal*. Property *is contained in* with subclasses: *hasAgriParcelParent*, *holdingPlot*, *holdingSite*, *holdingZone*, *Machine* property, *originType*, *soilProperty*, *Tractor* and *Within*.

The data properties that are used, are being described below: *Area* which shows the area of the parcel nominally in square meters, *Category* which shows the category of the parcel of land, if it is arable, grassland, vineyard, orchand, mixed crop, etc., *Code* (defined on Foodie), *createdAt*, *CropStatus* which describes the crop planting status such as seeded, justBorn, growing, maturing, readyForHarvesting, *Description* (defined by Foodie), *has serialization* which makes the connection between a geometry object with its text-based serialization, *hasName* (defined by Saref4agri), *lastPlantedAt* which indicates when the crop was last planted, *Notes* (defined by Foodie), *prov:generatedAtTime*, *prov:invalidatedAtTime*, *validFrom*, *validTo*.

D.4 Agriculture Crops ontology

The Agriculture Crops ontology is available under https://w3id.org/demeter/agri/agriCrop and is briefly presented below.

This entity contains a harmonised description of a generic crop. This entity is primarily associated with the agricultural vertical and related IoT applications.

There are three equivalent classes named *AnyFeature*, *Feature* and *FeatureType* that encapsulate the necessary subclasses enabling interoperability among existing ontologies. The aforementioned subclasses are *Crop* and *CropSpecies. Datatype* class has the *CropType* and *ProductionType* subclasses that define the types of crop holding specific cardinality restrictions in most properties. Note that instances of *CropType* can be linked to the concepts in AGROVOC using the *agroVocConcept* property from FIWARE. Other classifications may be also linked in future releases. *Geometry, measure, measurement, Property, PropertyType* are other classes defined in this model, used mostly for ranging purposes.

Object Properties are relationships defined between class objects. A number of such properties (*cropArea, cropHasAgriSoil,* cropSpecies, *hasAgriPest, hasAgriFertiliser, hasRank*) refer to crop features, while others to production (*production, productionAmount, productionProperty*). The rest deal with properties of crop (*has feature of interest, has property* and their inverse *is feature of interest of* and *is property of,* respectively).

Likewise, Data Properties match an object to a value and not another object. These properties are string-typed *family*, *code*, *genus*, *createdAt*. d*escription, harvestingInterval, name, notes, species, variety and* the datetime properties *plantingFrom, has plant date, validFrom* and *validTo,* the equivalent *productionDate* and *has harvest date,* as well the provenance ones (*generatedAtTime, invalidatedAtTime*)*.* There is also an enumeration, named *wateringFrequency*.

Figure 39. Visualization of the Agriculture Crops ontology

D.5 Agriculture Interventions ontology

The Agriculture Interventions ontology is available under https://w3id.org/demeter/agri/agriIntervention and is briefly presented below.

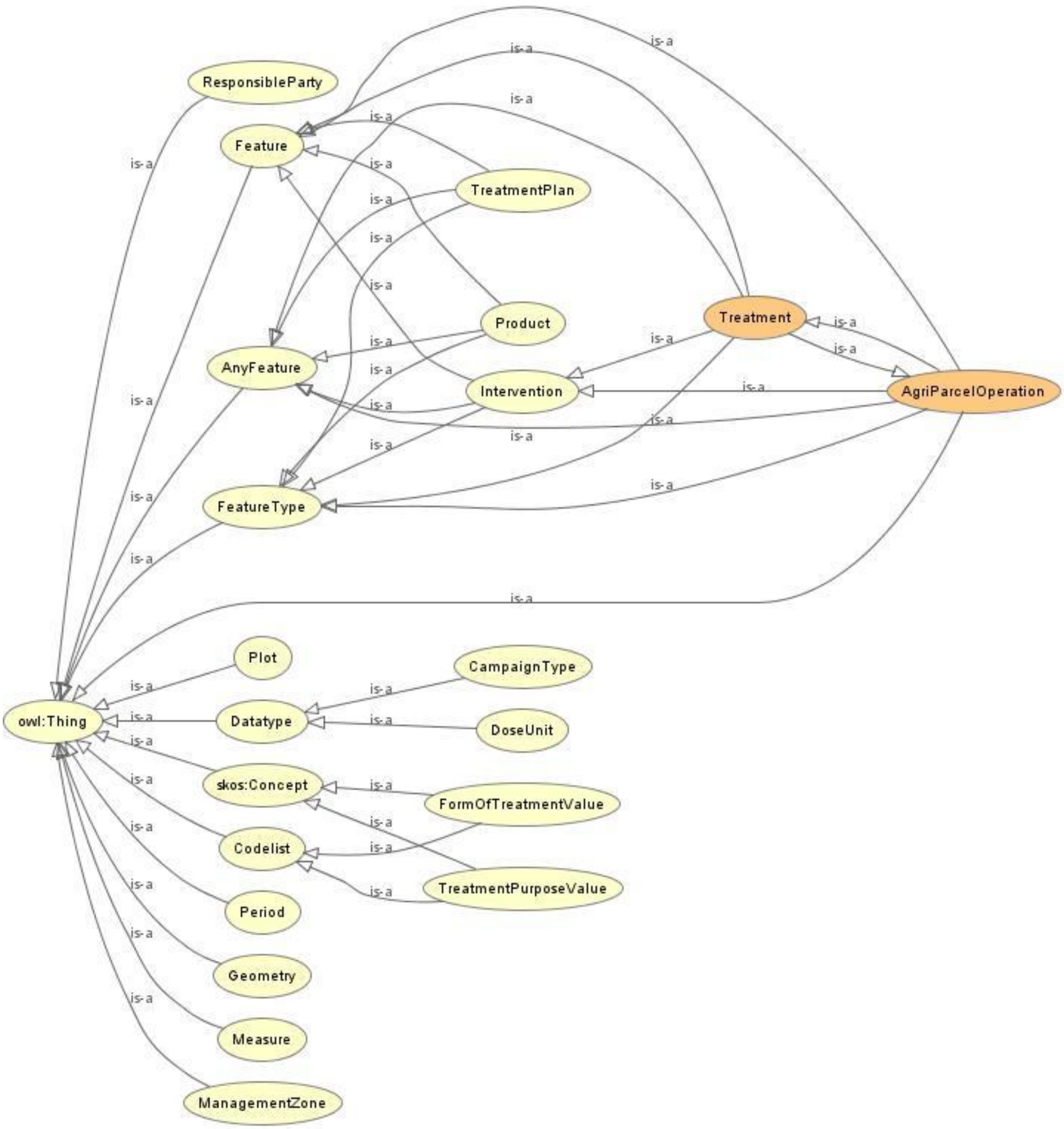

Figure 40. Visualization of the Agriculture Interventions ontology

This entity contains a harmonised description of generic operations performed on a parcel of land. This entity is primarily associated with the agricultural vertical and related IoT applications.

There are three equivalent classes named *AnyFeature*, *Feature* and *FeatureType* that encapsulate the necessary subclasses enabling interoperability among existing ontologies. The aforementioned subclasses are the equivalent *AgriParcelOperation* and *Treatment*, *TreatmentPlan* and *Product. Datatype* class has the *CampaignType* and *DoseUnit* subclasses that define a number of cardinality restrictions on properties, while the equivalent *Concept* and *Codelist* encapsulate the *FormOfTreatmentValue* and *TreatmentPurposeValue* subclasses. Note that *Product* may be connected also to AGROVOC concepts like pesticides types, fertilizers types, and in the future we may also connect other classifications. M*easure, Period, ResponsibleParty, ManagementZone, Geometry and Plot* are more classes defined in this model, used for ranging purposes.

Object Properties are relationships defined between class objects. This model basically matches the operation to the corresponding data, as defined by the classes discussed previously. The relations defined are *applicationWidth, campaign, evidentParty, flowAdjustment, formOfTreatment, interventionGeometry, interventionZone, maximumDose, minimumDose, motionSpeed, period, plan, planProduct, pressure, quantity* and *supervisor*. The rest deal with properties of the intervention (*hasOperator, hasAgriProductType, operationHasAgriParcel* and their equivalent *operator, treatmentProduct* and *interventionPlot,* respectively).

Likewise, Data Properties match an object to a value and not another object. These properties are the descriptive (literal or numerical) *price*, d*escription, quantity, treatmentDescription, treatmentPlanCode, treatmentPlanCreation and* the datetime properties *creationDateTime*. *reportedAt, validFrom* and *validTo, plannedStartAt* and *plannedEndAt.* There are also some enumerations (*result, status, operationType, waterSource*).

D.6 Agriculture Alerts ontology

The Agriculture Alerts ontology is available under https://w3id.org/demeter/agri/agriAlert and is briefly presented below.

The purpose of this model is to support the generation of notifications for a user or trigger other actions, based on alerts. An alert is generated by a specific situation. The main features of an alert are that it is not predictable and that it is not recurrent data. That means that an alert could be, for example, an accident or an extremely high level of measure.

There are three equivalent classes named *AnyFeature*, *Feature* and *FeatureType* that encapsulate the necessary subclasses enabling interoperability among existing ontologies. The aforementioned subclasses are *Alert*, *CropSpecies*, *ManagementZone* and *Plot*. *Geometry* is another class defined in this model, used for ranging purposes.

Object Properties are relationships defined between class objects. This model basically matches the alert to the corresponding data, as defined by the classes discussed previously. Consequently, the properties are *alertPlot, alertSpecies* and *alertZone* as well as the inverses *plotAlert, speciesAlert, zoneAlert* respectively. There is also a *location* property encapsulating the *alertGeometry* relationship that draws objects from Geometry class.

Likewise, Data Properties match an object to a value and not to another object. These properties are *address*, a*lertSource*, *code*, *data*, *dateIssued*, d*escription, severity, subCategory,* the equivalent *category* and *type* as well as *validFrom* and *validTo.*

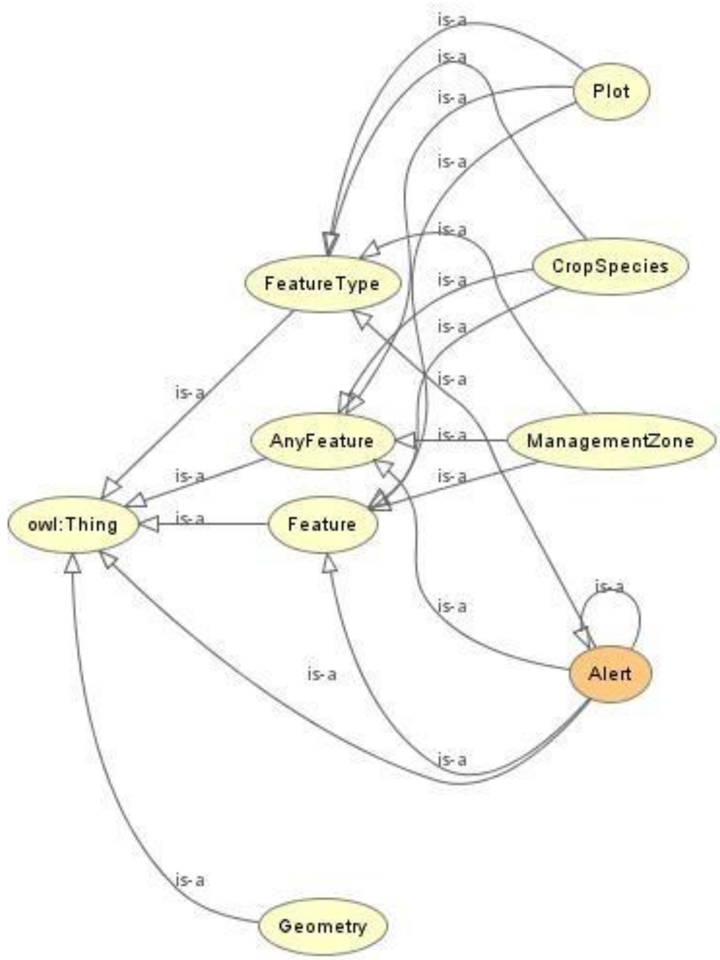

Figure 41. Visualization of the Agriculture Alerts ontology

D.7 Agriculture Product ontology

The Agriculture Product ontology is available under https://w3id.org/demeter/agri/agriProduct and is briefly presented below.

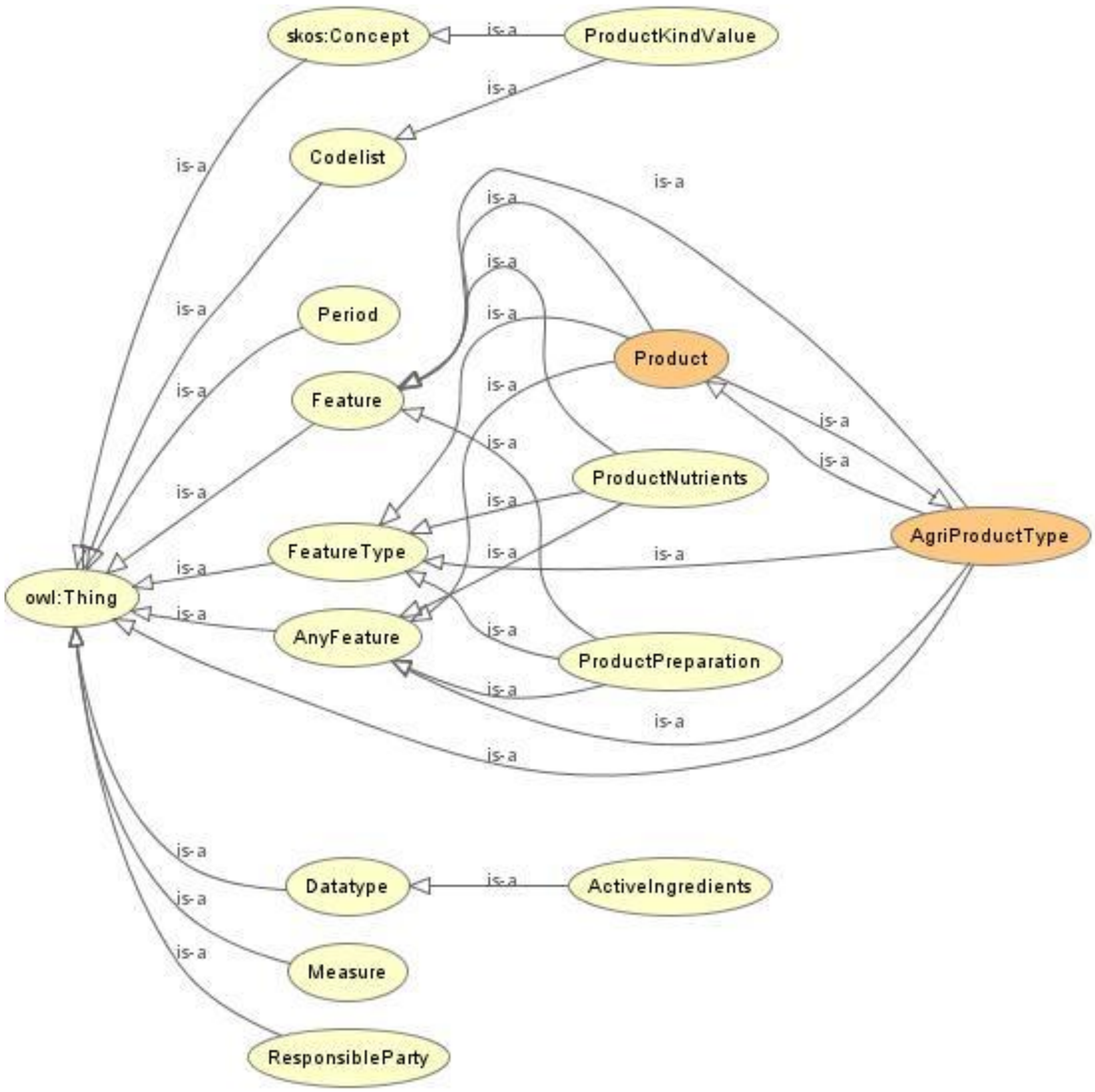

Figure 42. Visualization of the Agriculture Product ontology

This entity contains a harmonised description of a generic agricultural product type. This entity is primarily associated with the agricultural vertical and related IoT applications. The AgriProductType includes a hierarchical structure that allows product types to be grouped in a flexible way.

There are three equivalent classes named *AnyFeature*, *Feature* and *FeatureType* that encapsulate the necessary subclasses enabling interoperability among existing ontologies. The aforementioned subclasses are the equivalent *AgriProductType* and *Product*, *productNutrients* and *productPreparation. Datatype* class has the *activeIngredients* subclass and the equivalent *Concept* and *Codelist* encapsulate the *ProductKindValue* subclass. M*easure, Period, ResponsibleParty* are more classes defined in this model, used for ranging purposes.

Object Properties are relationships defined between class objects. This model basically matches the product to the corresponding data, as defined by the classes discussed previously. The relations defined are *ingredientAmount, manufacturer, nutrient, nutrientAmount, nutrientProduct, productKind, productQuantity, safetyPeriod, solventQuantity*. The rest deal with properties of the product (*hasAgriProductTypeChildren, hasAgriProductTypeParent*).

Likewise, Data Properties match an object to a value and not to another object. These properties are string-typed: *code*, d*escription, name (ingredientName, nutrientName, productName), productCode, productSubType, productType, registrationCode, safetyInstructions* and *storageHandling*, or numerical properties, such as: *nutrientMeasure* and *price,* logical *root* and the format-specific *registerUrl.*

D.8 Agriculture Properties ontology

The Agriculture Properties ontology is available under https://w3id.org/demeter/agri/agriProperty and is briefly presented below.

The agriProperty.ttl is consists of the following classes: *AgriParcelRecord* class which contains a harmonised description of the conditions recorded on a generic parcel of land. This entity is primarily associated with the agricultural vertical and related IoT applications. *Codelist*, which contains the subclass *PropertyTypeValue* , part of the foodie ontology. *Compressibility* class that contains the instance *vapour compressibility*, the latter contains information about the level of compression that a vapour has. *Concentration* class, which contains the instance carbon content. This indicates the carbon concentration on a farm field.

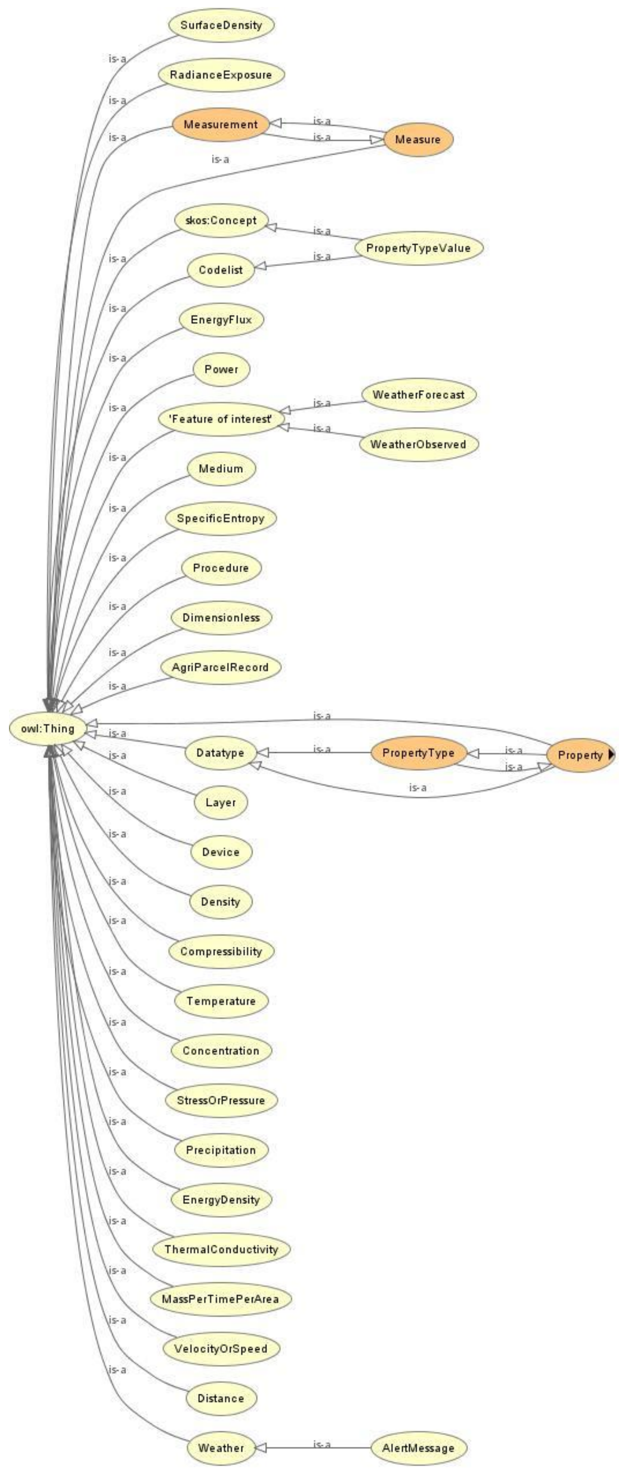

Figure 43. Visualization of the Agriculture Properties ontology

Datatype class. It contains the subclasses *Property* and *PropertyType*. *Property* class contains anything that can be sensed, measured or controlled in households, common public buildings or offices. We propose here a list of properties that are relevant for the purpose of Saref, but this list can be extended. These instances are the *Plant growth stage*, *Soil moisture*, *electricConductivity*, *pH*, *Precipitation*, *soilTexture*, *soilType*. *Property* class contains the *Humidity* class, responsible for the instance *Ambient humidity*, and the *Temperature* class, which is responsible for the instances: *Air temperature* and *Soil temperature*. *Density* class, with the instances *snow density* and *air density*, for measuring the snow density and air density in a field. *Device* class. *Dimensionless* class, with the instances *soil albedo*, *soil porosity* and *vegetation* area fraction. *Distance* class, with the instance *snow grain size*. *EnergyDensity* class, with the instance *sound energy density, EnergyFlux*, with the instances *sound intensity in air* and *sound intensity*. *Feature of interest* class, with subclasses *WeatherForecast* and *WeatherObserved*. *Layer* class, with the instances *soil layer* and *vegetation*. *Measure* class. *Measurement* class. *Medium* class with instances: *soil pores* and *soil*. *Power* class, with instance *sound power*. *Precipitation* class with instance *snowfall*. *Procedure* class. *Property* class with subclasses *Humidity*, Temperature and instances *Plant growth stage*, *Soil moisture*, *electricConductivity*, *pH*, *Preciptation, soilTexture*. *RadianceExposure* class. *skos:Concept* class with the subclass *PropertyTypeValue*. *SpecificEntropy* class with instance *soil thermal capacity*. *SurfaceDesnity class*, which consists of the instances *snow soot content*, *snowfall amount*, *atmosphere mass content of carbon dioxide, atmosphere*, *content of carbon monoxide*, *atmosphere water vapor content*, *soil frozen water content*, *soil moisture content at field capacity* and *vegetation carbon content*. *Temperature* class with *snow temperature* instance. *ThermalConductivity* class with the instance *soil thermal conductivity*.

The individuals that belong to the previous classes are the following:

For soil measurement, the agriProperty ontology uses the following individuals: *Soil*, *soil layer* , *Soil moisture*, *soil pores*, *Soil temperature*, *soil_albedo*, *soil carbon content*, *soil frozen water content*, "*soil hydraulic conductivity at saturation* , *soil moisture content at field capacity*, *soil porosity*, *soil suction at saturation*, *soil temperature*, *soil thermal capacity* , *soil thermal conductivity*, *soilTexture*, *soilType*, *volume fraction of clay in soil*, *volume fraction of condensed water in soil*, *volume fraction of condensed water in soil at critical point*, *volume fraction of condensed water in soil at field capacity*, *volume fraction of condensed water in soil at wilting point*, *volume fraction of condensed water in soil pores*, *volume fraction of frozen water in soil*, *volume fraction of sand in soil*, *volume fraction of silt in soil*, *moisture content of soil layer*, *moisture content of soil layer at field capacity*, *downward heat flux in soil*, *lwe thickness of soil moisture content*, *pH*, *electricConductivity*.

As far as the atmosphere and air are concerned, the following individuals are used: *air temperature*, that measures the degree of intensity of heat present in the air, *air density*, like air pressure, decreases with increasing altitude. It also changes with variation in atmospheric pressure, temperature and humidity, *ambient humidity*, which shows the amount of water vapour in the air, *atmosphere mass content of carbon dioxide, atmosphere mass content of carbon monoxide*, a*tmosphere water vapor content*, *degree Celsius*, *Millibar*.

As far as the snow/water are concerned, we use the individuals: *lwe convective snowfall rate*, *lwe large scale snowfall rate*, *lwe snowfall rate*, *lwe thickness of convective snowfall amount*, *lwe thickness of frozen water content of soil layer*, *lwe thickness of large scale snowfall amount*, *lwe thickness of moisture content of soil layer*, *lwe thickness of snowfall amount*, *snow density*, *snow grain size*, *snow soot content*, *snow temperature*, *snowfall*, *snowfall amount*, *snowfall flux*, *frozen water content of soil layer*, *water*, *water evaporation flux from soil*, *convective snowfall amount*, *convective snowfall flux*, *mass concentration of condensed water in soil*, *thickness of convective snowfall amount*, *thickness of large scale snowfall amount*, *thickness of snowfall amount*, *large_scale_snowfall_amount*, *large scale snowfall flux* , *liquid water content of soil layer*, *precipitation*, *Millimetre*.

As far as the environmental noise is concerned, we use the individuals: *sound pressure in air*, *Compressibility*, *Pressure*, *decibel-milliwatts*, *Millivolt*. *Power* of noise.

Relative to the vegetation, the following individuals are used: *normalized difference vegetation index* (NDVI) which is a graphical indicator in order to analyze remote sensing measurements usually from space platform and can identify whether the target contains live green vegetation*, vectorProperty*, *Vegetation*, *vegetation area fraction*, *vegetation carbon content*, *carbon content*, *Plant growth stage*.

Concerning the Object properties of this ontology, the following are used: *controls property*, *generalQuantityKind*, *has feature of interest*. The property *has property* with subclasses *productionProperty* and *soilProperty*. The property *hasDevice* and its equivalent *devices, smartMeter*, *hasProperty*, *is controlled by device*, *is measured by device*, *is measured in*, *is property of*, *isFeatureOfInterestOf*, *IsPropertyOf*, *makes measurement*, *measurement made by*, *measures property*, *propertyType*, *propertyType*, *recordHasAgriParcel*, *refDevice*, *refPointOfInterest*. The property *relates to measurement* with subclasses: *productionAmount*, *quantitativeProperty*. And, finally, the *relates to property.*

The data properties that are used, are being described below: *analysisDate* (defined by Foodie), *dateObserved* which contains the date and time of this observation in ISO8601 UTCformat. It can be represented by an specific time instant or by an ISO8601 interval, *has timestamp* (defined by Saref), *has value* (defined by Saref) , with the following subclasses: *airTemperature*, which is the observed air temperature (in the shade) nominally in degrees centigrade, *atmosphericPressure* (defined by Fiware) which shows the atmospheric pressure observed and measured in Hecto Pascals, *dewPoint* (defined by Fiware) where the dew point encoded as a number, *Illuminance* (defined by Fiware) which stores the illuminance observed measured in lux (lx) or lumens per square metre (cd·sr·m−2), *nonQuantitativeProperty* (defined by Foodie), *pressureTendency* (defined by Fiware) which expresses the rising or falling pressure in quantitative terms or qualitative terms in values rising, falling or steady, *relativeHumidity* (defined by Fiware) which contains air's relative humidity observed, *snowHeight* (defined by Fiware) which displays the snow height observed by generic snow depth measurement sensors, expressed in centimeters. *soilMoistureEc* which is measured as Electrical Conductivity, in units of Siemens per meter (S/m), *soilMoistureVwc* which is measured as Volumetric Water Content, VWC as a percentage. 0 ≤ soilMoistureVwc ≤ 1, *soilTemperature* which is the observed soil temperature nominally in degrees centigrade, *Temperature* (defined by Fiware) which stores the observed air's temperature, *Visibility* (defined by Fiware) which contains the visibility reported as veryPoor, poor, moderate, good, veryGood, excellent, *windDirection* (defined by Fiware) which contains the wind direction expressed in decimal degrees compared to geographic North (measured clockwise), encoded as a Number. Range 0 to 360. *windSpeed* (defined by Fiware). Next is the property *Name* with subclass *propertyName* (defined by Foodie). There is the *observedAt* which indicates the time and the date that the record was observed, *weatherType* (defined by Fiware) which is the observed weather type. It is represented by a comma separated list of weather statuses, for instance overcast, lightRain. Finally, *salinity* describes the salt content of a body of water usually measured in g/mL and *totalConsumption* expresses the energy consumed by any device in kilowatt-hours.

D.9 Agriculture Systems ontology

The Agriculture Systems ontology is available under https://w3id.org/demeter/agri/agriSystem and is briefly presented below.

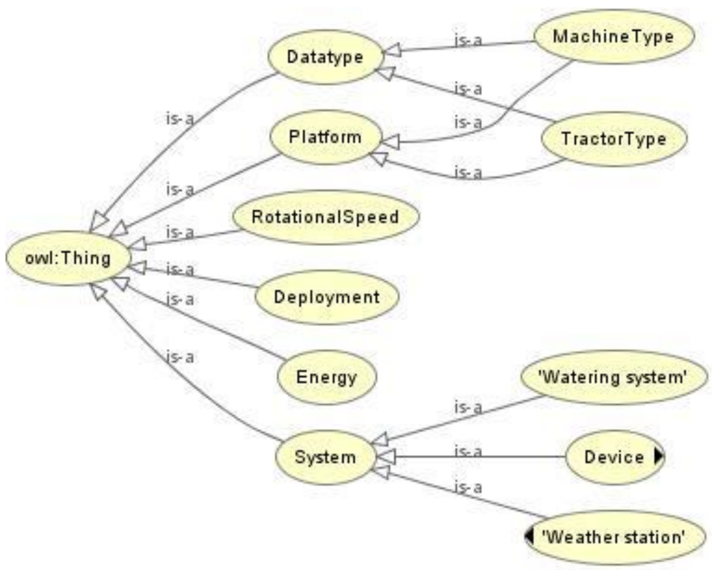

Figure 44. Visualization of the Agriculture Systems ontology

The AgriSystem.ttl ontology consists of the following classes: the class Datatype, which contains the class *MachineType* and the *TractorType* class, both of them defined by the Foodie ontology. *Deployment* class is defined by SSN ontology. *Energy* class. *Platform* class (defined by SOSA ontology) that contains the subclasses *MachineType* (defined by Foodie ontology) and *TractorType* (defined by Foodie ontology). Platform entity hosts other entities, such as Sensors, Actuators, Samplers and other Platforms. *RotationalSpeed* class, with the instances *critical build-up speed*, *critical torsional speed*, *critical whirling speed*, and *synchronous speed*. *System* class (defined by SSN ontology), which contains pieces of infrastructure that implement Procedures. Subclass of *System* class is the *Device* class (defined by SAREF ontology). The latter ontology contains the *Actuator* class (defined by SAREF ontology) and the *Sensor* class (defined by SAREF ontology). *Actuator* class contains the following subclasses: *Watering gun* class (defined by Saref4agri), which is an actuator to irrigate a space and *Watering valve* class (defined by Saref4agri). *Sensor* class contains the classes: *Eating activity sensor* (defined by Saref4agri), *Milking sensor* (defined by Saref4agri), *Movement activity sensor* class (defined by Saref4agri), *Pluviometer* class (defined by Saref4agri), which is a sensor for measuring the rain fall, *Soil tensiometer* class (defined by Saref4agri) which is a sensor for measuring the soil moisture, *Thermometer* class (defined by Saref4agri), *Weather station* class (defined by Saref4agri) which is a sensor or a system for measuring weather conditions, *Weight sensor* class (defined by Saref4agri). And finally, *Watering station* class (defined by Saref4agri).

It also uses the *code* data property.

D.10Agriculture Pests ontology

The Agriculture Pests ontology is available under https://w3id.org/demeter/agri/agriPest and is briefly presented below.

Figure 45. Visualization of the Agriculture Pests ontology

AgriPest.ttl ontology consists of the *agriPest* class. This class describes the agricultural pest. It is primarily associated with the agricultural vertical and related IoT applications. It has only one object property, the *hasAgriProductType* which is a reference to recommended types of products that can be used to treat this pest. This ontology contains the following 3 data properties: *alternateName*, *description* and *name*. Additionally, note that using the *agroVocConcept* property (from the common module), individuals of the *agriPest* class can be connected to the equivalent pest concept from AGROVOC.

D.11Farm Animals ontology

The Farm Animals ontology is available under https://w3id.org/demeter/agri/farmAnimal and is presented in the figure below.

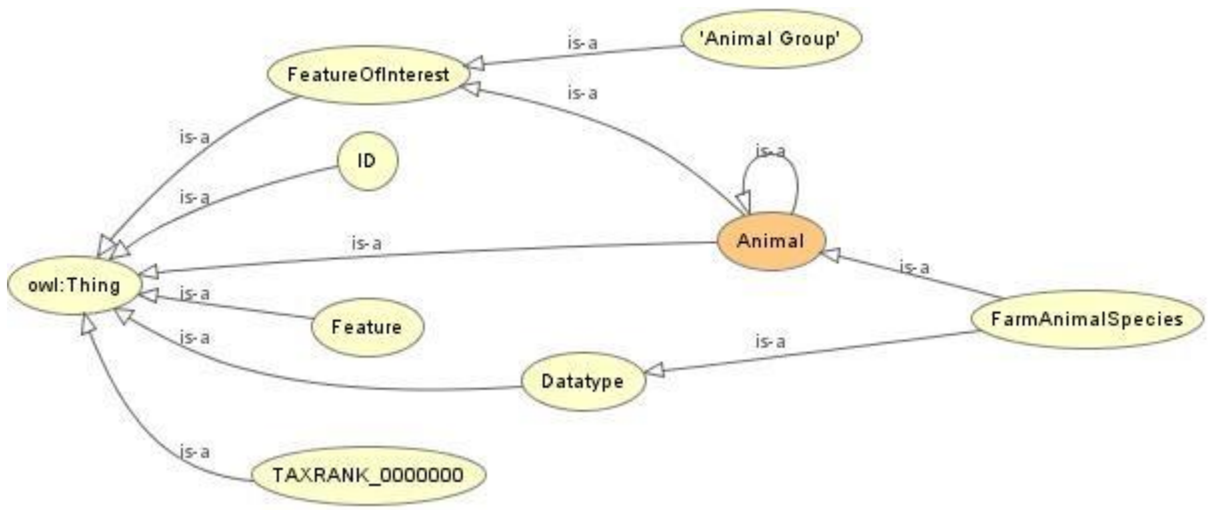

Figure 46. Visualization of the Farm Animals ontology

This module describes the proposed animal data model that has been made from a more general point of view, trying to adjust it to the information coming from the devices and sensors used to monitor or record the animals, their status, their relationships and properties in general.

The class *Datatype* encapsulates the *FarmAnimalSpecies* subclass, which is also a subclass of *Animal* and defines every possible type of animal we might encounter in our data. This subclass defines a number of restrictions that will be explained in the next steps. The aforementioned hierarchy is expressed in *FeatureOfInterest* class which also encapsulates the *Animal Group* subclass which is a collection of animals. Other classes are *ID* and *TAXRANK_0000000* which are necessary for identifying and telling the animals apart.

Object Properties are relationships defined between class objects. Such properties could define relationships between different animals such as parenthood (*calvedBy*, *siredBy*) or between animal and person (*ownedBy*). Other relationships defined between objects of the animal subclasses are *has member* and its inverse *is member of*, *has id*, *includesAnimal*. There are also location related properties like *is located in* and its subclass *locatedAt* and the inverse *is location of*.

Likewise, Data Properties match an object to a value and not to another object. Some of these properties should be unique for each object such as *birthdate*, *has birth date*, *hasName*, *livestockNumber*, *livestockType*. Other just describe the animal like *breed*, *sex*, *weight*, *species* and most of the rest are related to the condition of the animals. These are *healthCondition*, *phenologicalCondition*, *reproductiveCondition*, *welfareCondition*. Finally, there are some other properties like *legalID* and *relatedSource*, which is the ID used for the animal in external applications.